

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



# Novas funcionalidades para electrodomésticos de cozinha num cenário de Internet-das-coisas

João Amorim Baixinho

RETROFITTING OF LEGACY KITCHEN APPLIANCES FOR AN IOT  
SCENARIO

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: João Paulo de Sousa

31 de Julho de 2018



# Resumo

Nos dias de hoje observamos uma constante evolução tecnológica e uma diminuição dos custos de produção dos componentes eletrônicos, fazendo com que os fabricantes de eletrodomésticos apostem no lançamento de novos modelos para o mercado, cada vez mais evoluídos do ponto de vista da eletrônica de controlo.

Esta evolução faz com que os eletrodomésticos tenham a possibilidade de aumentar o número de funcionalidades de tal forma que hoje em dia se fala no conceito de interligar os eletrodomésticos entre si e a outros objetos (Internet das coisas), fazendo com que possam ser controlados através de dispositivos remotos, como telemóveis, tablets ou computadores.

Após se ter efetuado um estudo alargado no mercado de máquinas de lavar loiça, concluiu-se haver espaço para se desenvolver um produto capaz de tornar conectáveis os modelos de gama média e baixa que atualmente, por razões de estratégia comercial, ainda não possuem essa funcionalidade.

Desenvolveu-se assim, como prova de conceito, um controlador eletrónico capaz de ser utilizado em máquinas de lavar loiça simples. Este controlador eletrónico substitui o controlador eletromecânico existente, recebendo informação de todos os sensores que a máquina possui de forma a saber o estado em que esta se encontra, para depois decidir como controlar os componentes internos da máquina.

O controlador eletrónico desenvolvido, para além de reproduzir o ciclo de lavagem que o controlador eletromecânico produz, oferece funcionalidades de valor acrescentado. Estas funcionalidades são a possibilidade de informar o utilizador, através de uma aplicação *android* que poderá ser instalada em qualquer smartphone ou tablet, do valor da temperatura, do consumo de água, do consumo de energia, do custo de lavagem, bem como a possibilidade de replicar remotamente todas as ações de comando que se podem efetuar no painel de controlo da máquina.



# Abstract

Nowadays we observe a steady technological evolution, which leads, year after year, to the decrease of the production costs associated to electronic components. Following this tendency, most manufactures of household appliances are launching more and more sophisticated models of their products.

This continued evolution led to the possibility of expanding the number of features, most notably the capacity of interconnecting different household appliances among them and to other objects (Internet of Things) making them, among other possible scenarios, controllable by remote devices such as mobile phones, tablets or computers.

This work started by conducting a market research with the objective of better understanding whether or not there was space to develop a flexible controller capable of replacing the electrical electromechanical controller and providing additional features.

The product developed as proof of concept was a flexible electronic controller, capable of being retrofitted to classical (electro mechanically controlled) dishwashers. This electronic controller replaces the electromechanical controller of the appliance, receives information from all the sensors and controls all the components of the dishwasher.

In addition to replicating the washing cycle produced by the electromechanical controller, this product offers new features, such as informing the user, through a mobile application, of the values of different parameters, including the washing temperature, the water and energy consumption and the total cost of the washing cycle. It is also possible to change the characteristics of the washing cycle and to connect the machine to a fixed or mobile device for remote operation and monitoring.



# Agradecimentos

Ao longo da meu percurso académico tive o prazer de conhecer muitas pessoas que me apoiaram e contribuíram de várias formas para minha evolução enquanto estudante.

Uma destas pessoas é o professor João Paulo de Sousa, orientador da minha dissertação, que se mostrou muito empenhado desde o início desta ultima fase da minha formação académica. Agradeço desde já o interesse em partilhar conhecimento e experiência e por me acompanhar neste desafio.

Com pena minha apenas no terceiro ano pude conhecer o meu colega e amigo Artur Antunes que me ajudou a estimular o meu interesse por aprender e por partilhar o seu conhecimento durante esta fase.

À FEUP e a todos os docentes, engenheiros, técnicos que de uma forma ou de outra despenderam algum do seu tempo em partilhar o seu saber, experiência e que de algum modo contribuíram para a minha formação.

À minha namorada Joana Rodrigues que esteve sempre presente nos momentos mais complicados e sempre disposta em ajudar naquilo que fosse preciso.

Agradeço a todos os familiares e amigos que me deram força durante este cinco anos. Um agradecimento especial à minha irmã pelo incentivo e apoio dado estes anos.

À minha mãe que fez tudo o que estava ao seu alcance para que tivesse uma formação académica com todas as condições necessárias e que me apoiou em todas as decisões que tomei durante estes cinco anos.

João Baixinho





*“It never gets easier, you just get faster  
- Greg LeMond”*



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Motivação e Objetivos . . . . .	2
1.3	Definição do problema . . . . .	2
1.4	Guia de leitura . . . . .	2
<b>2</b>	<b>Estudos de mercado</b>	<b>3</b>
2.1	Opções de compra para máquinas de lavar loiça . . . . .	3
2.2	Soluções para o dispositivo de controlo remoto . . . . .	4
<b>3</b>	<b>Solução proposta</b>	<b>7</b>
3.1	Características da Solução . . . . .	7
3.2	Organização do trabalho . . . . .	8
3.2.1	Instalação laboratorial . . . . .	8
3.2.2	Instrumentação da máquina . . . . .	9
3.2.3	Caracterização do ciclo de lavagem . . . . .	9
3.2.4	Projeto do controlador . . . . .	9
3.2.5	Ensaio preliminares e validação de resultados . . . . .	9
3.2.6	Protótipo industrial . . . . .	9
3.2.7	Desenvolvimento de funcionalidades adicionais . . . . .	10
3.2.8	Acesso remoto . . . . .	10
3.2.9	Gestão do projeto . . . . .	10
3.3	Diagramas de Gantt . . . . .	10
<b>4</b>	<b>Caracterização da máquina e de um ciclo de lavagem</b>	<b>13</b>
4.1	Funcionamento geral de uma máquina de lavar loiça . . . . .	13
4.1.1	Ciclo de lavagem de uma máquina simples . . . . .	14
4.1.2	Necessidade do uso de detergente e sal . . . . .	14
4.1.3	Abrilhantador . . . . .	15
4.2	Caracterização do ciclo de lavagem . . . . .	16
4.3	Componentes internos . . . . .	16
4.3.1	Bomba de circulação . . . . .	17
4.3.2	Bomba de extração . . . . .	18
4.3.3	Resistência de aquecimento . . . . .	18
4.3.4	Circuito de acondicionamento da água de lavagem . . . . .	18

4.3.5	Dispensador de detergente e líquido abrillantador . . . . .	19
4.3.6	Termostatos . . . . .	19
4.3.7	Ciclo de lavagem de máquina Balay V4521 . . . . .	19
4.3.8	Diagrama de estados da máquina . . . . .	20
<b>5</b>	<b>Instalação e instrumentação da máquina</b>	<b>23</b>
5.1	Preparação do laboratório e instalação da máquina . . . . .	23
5.2	Instrumentação da máquina . . . . .	24
5.2.1	Software para recolha de dados . . . . .	25
5.3	Calibração do caudalímetro . . . . .	26
<b>6</b>	<b>O novo controlador da máquina</b>	<b>27</b>
6.1	Cablagem e diagrama de blocos . . . . .	27
6.2	Requisitos . . . . .	28
6.3	Módulo de controlo . . . . .	29
6.4	Módulo de potência . . . . .	30
6.5	Alterações posteriores . . . . .	33
<b>7</b>	<b>Interface com o utilizador</b>	<b>35</b>
7.1	Interface local . . . . .	35
7.2	Aplicação para interface remota . . . . .	37
7.3	Comunicação microcontrolador - aplicação Android . . . . .	41
7.4	Ferramenta de desenvolvimento . . . . .	42
7.5	Desenvolvimento da aplicação . . . . .	43
<b>8</b>	<b>Conclusão</b>	<b>47</b>
8.1	Resultados e conclusões . . . . .	47
8.2	Trabalho futuro . . . . .	48
	<b>Referências</b>	<b>49</b>
<b>A</b>	<b>Código do Datalogger</b>	<b>51</b>
<b>B</b>	<b>Código do Controlador</b>	<b>55</b>

# Lista de Figuras

3.1	Sucessão temporal do plano de trabalho . . . . .	8
3.2	Diagrama de Gantt - 1ª fase . . . . .	11
3.3	Diagrama de Gantt - 2ª fase . . . . .	12
4.1	Composição de Detergente . . . . .	15
4.2	Máquina de lavar loiça . . . . .	17
4.3	Motor de polos sombreados . . . . .	18
4.4	Pré lavagem . . . . .	20
4.5	lavagem completa . . . . .	21
4.6	microciclo . . . . .	22
4.7	microciclo . . . . .	22
5.1	Esquema elétrico original da máquina . . . . .	24
5.2	Circuito de isolamento . . . . .	25
5.3	Circuito de isolamento . . . . .	25
6.1	Cablagem e diagrama de blocos da máquina . . . . .	28
6.2	Esquemático da placa de controlo . . . . .	30
6.3	Placa de circuito impresso do módulo de controlo . . . . .	31
6.4	Circuito isolador . . . . .	31
6.5	Esquemático da placa de potência . . . . .	32
6.6	Placa de circuito impresso do módulo de potência . . . . .	33
6.7	Circuito de alimentação do LED5 . . . . .	34
6.8	Circuito de correção do Output Enable . . . . .	34
7.1	Painel de controlo da máquina . . . . .	35
7.2	Visão geral . . . . .	39
7.3	Menu principal . . . . .	39
7.4	Características de programa . . . . .	39
7.5	Estado da máquina . . . . .	39
7.6	Modo técnico . . . . .	39
7.7	Painel de desenho dos ecrã . . . . .	42
7.8	Painel da programação em blocos . . . . .	44



# Lista de Tabelas

2.1	Máquinas de lavar loiça disponíveis em 2017 e 2018 . . . . .	4
2.2	Telemóveis disponíveis em 2017 e 2018 . . . . .	5
3.1	Lista das Workpackages . . . . .	8
4.1	Componentes atuadores da máquina Balay V4521 . . . . .	17
5.1	Dados para calibração do caudalímetro . . . . .	26
5.2	Volume de água (litros) e impulsos por litro . . . . .	26
6.1	Ligações ao Arduíno Nano . . . . .	30
7.1	Narrativas de utilização . . . . .	38
7.2	Casos de uso . . . . .	38
7.3	Comandos enviados pelo cliente . . . . .	41





# Abreviaturas e Símbolos

IoT	Internet of Things - Internet das Coisas, Internet dos objetos
WP	Workpackage
SF	Shift-register
GUI	Grafical user interface



# Capítulo 1

## Introdução

### Conteúdo

1.1	Contexto . . . . .	1
1.2	Motivação e Objetivos . . . . .	2
1.3	Definição do problema . . . . .	2
1.4	Guia de leitura . . . . .	2

### 1.1 Contexto

Com o aumento da disponibilidade de largura de banda, a diminuição do custo de cada conexão, o aumento do número de dispositivos que possuem recursos de comunicação, sensores e atuadores, e com a diminuição do custo destes dispositivos, reuniram-se as condições necessárias para se começar a pensar na Internet das Coisas (IoT - Internet of Things). A IoT é uma realidade do dia a dia em que vivemos, sendo um dos resultados da evolução tecnológica a que assistimos.

A IoT assenta num conceito de interligação de dispositivos físicos e objetos (veículos, eletrodomésticos, aparelhos eletrônicos em geral dispendo de sensores, atuadores e hardware e software de comunicação). Ao estarem interligados, os objetos passam a poder trocar mensagens com informações sobre o seu estado, abrindo a porta a cenários de interação nunca antes imaginados. Um dos cenários mais simples é a possibilidade de um utilizador poder controlar os objetos remotamente através por exemplo de uma aplicação móvel. Para isso apenas é necessário que a aplicação seja capaz de interrogar os sensores existentes num dado objeto ligado a uma infra-estrutura de comunicação, processar essa informação e dar ordens aos atuadores desse objeto por forma a controlar o seu funcionamento. Esta infra-estrutura de comunicação é habitualmente uma rede de comunicação baseada em

tecnologias sem fios, em particular Wi-Fi e Bluetooth. Uma vantagem deste tipo de infraestrutura é que é facilmente escalável, ou seja, é fácil adicionar mais objetos a uma rede já existente.

## 1.2 Motivação e Objetivos

Atualmente o mercado global oferece um gama muito grande de eletrodomésticos com a capacidade de serem controlados remotamente. Porém se restringirmos o mercado apenas à realidade portuguesa, as opções são mais reduzidas e, normalmente, apenas disponíveis em modelos de gama alta. Existe pois espaço de mercado para desenvolver um controlador flexível, capaz de ser incorporado em produtos de gama média e baixa. Foi esta, portanto, a motivação para este trabalho.

## 1.3 Definição do problema

O desafio aceite pode ser definido da seguinte forma: *desenvolver um controlador eletrónico para ser incorporado em máquinas de lavar loiça de gama média e baixa que substitua por completo o controlador eletromecânico existente e tenha funcionalidades adicionais, de elevado valor acrescentado, nomeadamente a possibilidade de conexão remota.* A restrição a máquinas de lavar loiça decorre de limitações logísticas e operacionais incontornáveis no período de tempo disponível.

A tese que neste trabalho se defende é que é possível desenvolver um tal controlador recorrendo apenas a componentes eletrónicos facilmente acessíveis (COTS - Components Off The-Shelf).

## 1.4 Guia de leitura

Este relatório está dividido em 7 capítulos. Começa-se com um breve estudo de mercado, onde estão descritos quatro estudos, dois relativos às opções de compra para máquinas de lavar loiça e outros dois relativos às soluções para o dispositivo de controlo remoto. De seguida descreve-se a solução proposta para o desenvolvimento do projeto e do planeamento do mesmo. Passando para a caracterização da máquina. Depois temos o capítulo que descreve a instalação e instrumentação da máquina.

Passa-se para o capítulo em que é descrito o desenvolvimento do novo controlador eletrónico da máquina. No seguinte capítulo é descrita a interface com o utilizador da máquina. Por último temos a resultados e conclusões e trabalho futuro. Quanto aos anexos, estes são dois. O anexo A contém o código do Datalogger e o B contém o código do controlador eletrónico.

## Capítulo 2

# Estudos de mercado

Neste capítulo apresentamos os dois estudos de mercado realizados. Restringimos à realidade Portuguesa. Por motivos logísticos e operacionais, escolheu-se como alvo o mercado de máquinas de lavar loiça e de telemóveis *Android*.

### Conteúdo

---

2.1	Opções de compra para máquinas de lavar loiça . . . . .	3
2.2	Soluções para o dispositivo de controlo remoto . . . . .	4

---

## 2.1 Opções de compra para máquinas de lavar loiça

Hoje em dia são diversos os fabricantes que produzem máquinas de lavar loiça conectáveis, porém no mercado português a realidade é bem diferente - em Março de 2017 fez-se um estudo com base em dados disponibilizados pela DECO<sup>1</sup> e pelos sites dos fabricantes onde foram analisados 139 modelos de máquinas de lavar loiça de 25 marcas diferentes [1].

Esse estudo assentou na contabilização dos modelos oferecidos por cada fabricante, quantos de entre eles possuíam algum tipo de conectividade e qual o preço mínimo a que podiam ser adquiridos. A tabela 2.1 apresenta os resultados desse estudo. Para cada marca apresentam-se os valores em Março de 2017 e a sua atualização em Junho de 2018.

Verifica-se que em 2017, das 25 marcas analisadas apenas uma possuía modelos com a capacidade de serem controlados remotamente. Em 2018 o número de modelos conectáveis aumentou mas confirma-se uma tendência que só se consegue explicar por razões comerciais: são apenas os modelos topo de gama a possuir possibilidade de conexão remota.

---

<sup>1</sup>DECO - Associação Portuguesa para a Defesa do Consumidor

Tabela 2.1: Máquinas de lavar loiça disponíveis em 2017 e 2018

Marca	Modelos não conectáveis	Preço (Euros)	Modelos conectáveis	Preço (Euros)
AEG	13/16	330/362	0/0	
BALAY	13/14	247/255	0/0	
BECKEN	1/1	250/400	0/0	
BEKO	4/4	212/209	0/0	
BOSCH	8/10	296/293	0/0	
CANDY	9/9	230/225	3/6	403/422
ELECTROLUX	10/9	317/304	0/0	
HOOVER	10/2	282/399	0/2	-/399
HOTPOINT-ARISTON	8/7	258/258	0/0	
INDESIT	4/6	235/237	0/0	
JBC	1/-	199/-	0/-	
JOCEL	1/1	222/218	0/0	
KUNFT	1/1	229/240	0/0	
LG	2/2	433/461	0/0	
MEIRELES	2/2	203/220	0/0	
MIELE	7/30	525/550	0/4	-/1550
ORIMA	4/4	201/205	0/0	
PRINCESS	2/2	250/210	0/0	
QILIVE	2/-	269/-	0/-	
SELECLINE	3/-	239-	0/-	
SIEMENS	11/15	314/396	0/2	-/874
TEKA	8/10	240/232	0/0	
TELEFUNKEN	1/-	223/-	0/-	
WHIRPOOL	10/9	259/239	0/0	
ZANUSSI	4/4	265/275	0/0	

## 2.2 Soluções para o dispositivo de controlo remoto

Às mesmas datas foi realizado um outro estudo de mercado, desta vez para conhecer as soluções de baixo custo que permitissem controlar remotamente a máquina de lavar. Este segundo estudo baseou-se também nos dados disponibilizados pela DECO relativos a telemóveis. Por motivos logísticos e operacionais consideraram-se apenas telemóveis baseados no sistema operativo *Android*. A base de dados disponibilizada possuía cerca de 200 modelos de telemóveis diferentes, com o seu respetivo preço e características [2]. Eram requisitos que o telemóvel possuísse tecnologia Wifi e Bluetooth 4.0 ou mais recente, e custo inferior a cem euros (um quarto do valor de uma máquina de lavar loiça conectável).

A tabela 2.2 resume os resultados desse estudo, realizado em Março de 2017 e atualizado em Junho de 2018.

Tabela 2.2: Telemóveis disponíveis em 2017 e 2018

Marca	Número de modelos 2017/2018	Preço (Euros) 2017/2018
ALCATEL	3/7	63/48
ACER	1/0	55/-
ALTICE	0/1	-/100
ASUS	0/1	-/85
BQ	1/0	99/-
HUAWEI	3/1	54/100
LAIQ	0/1	-/100
LENOVO	1/1	99/74
LG	2/3	89/79
MICROSOFT	1/0	79/-
NOKIA	0/1	-/100
NOS	0/1	-/75
SAMSUNG	1/2	83/95
WIKO	3/8	49/50
VODAFONE	1/1	45/90
XIAOMI	0/1	-/90

Com base na pesquisa efetuada pode concluir-se que no mercado Português existem inúmeras opções compra, de baixo custo, que satisfazem as necessidades de comunicação deste trabalho.





## Capítulo 3

# Solução proposta

Depois de se conhecer o mercado português de máquinas de lavar loiça e telemóveis Android de baixo custo apresenta-se neste capítulo a solução que se propõe para o problema a resolver.

### Conteúdo

<b>3.1</b>	<b>Características da Solução . . . . .</b>	<b>7</b>
<b>3.2</b>	<b>Organização do trabalho . . . . .</b>	<b>8</b>
<b>3.3</b>	<b>Diagramas de Gantt . . . . .</b>	<b>10</b>

### 3.1 Características da Solução

A substituição do controlador eletromecânico existente por um controlador eletrónico, leva a que o novo controlador tenha de ser capaz de reproduzir fielmente o trabalho executado pelo controlador antigo. Para que fosse possível esta substituição foi necessário conhecer quais são os componentes presentes numa máquina típica que necessitam ser controlados (atuadores) e quais os componentes necessários para conhecer o estado em que o ciclo de lavagem se encontra (sensores).

O novo controlador, para além de reproduzir o ciclo de lavagem já existente, fornece informações sobre o estado interno da máquina e das variáveis do processo.

Para medir a qualidade do ciclo de lavagem foi medido não só o consumo de água mas também o consumo de energia. Com estas duas variáveis pode calcular-se o custo total de um ciclo de lavagem, aplicando as tarifas praticadas pelos distribuidores de energia e de água.

O novo controlador possui uma interface simples com o utilizador, constituída por LEDs e um botão de comando rotativo. Também oferece possibilidade de comunicar com o exterior

de forma a que o utilizador possa definir remotamente os parâmetros de lavagem e receber informações sobre o estado da máquina, durante o ciclo de lavagem.

## 3.2 Organização do trabalho

Com o objetivo de organizar o calendário de trabalho, dividiu-se o projeto em workpackages e cada uma delas em tarefas. A tabela 3.1 apresenta lista das workpackages e a figura 3.1 a sua sequência temporal. Segue-se uma apresentação detalhada de cada workpackage.

Tabela 3.1: Lista das Workpackages

	Nome	Início	Fim
WP1	Instalação no laboratorial	06-03-17	20-03-17
WP2	Instrumentação da máquina	20-03-17	02-04-17
WP3	Caracterização do ciclo de lavagem	03-04-17	3-04-17
WP4	Projeto do controlador	01-05-17	20-05-17
WP5	Ensaio preliminares e validação de resultados	05-02-18	11-02-18
WP6	Acesso remoto	12-02-18	14-04-18
WP7	Protótipo industrial	15-05-18	16-06-18
WP8	Funcionalidades adicionais	15-05-18	16-06-18
WP9	Gestão do projeto	20-02-17	25-06-18

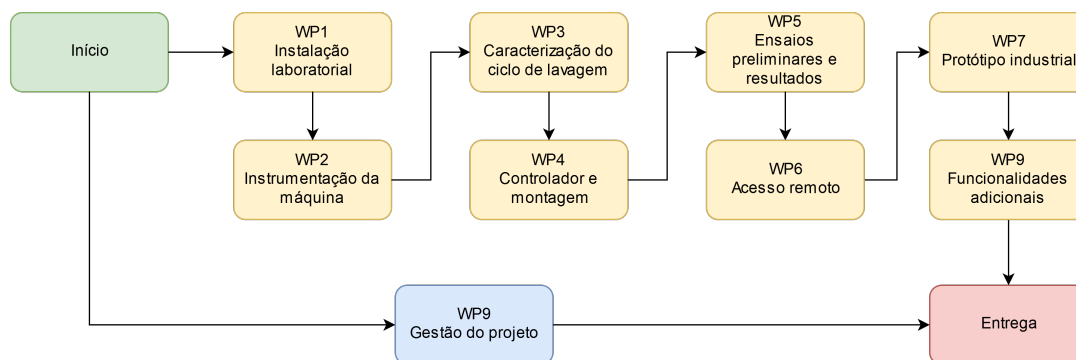


Figura 3.1: Sucessão temporal do plano de trabalho

### 3.2.1 Instalação laboratorial

O objetivo desta workpackage foi preparar o local de trabalho, começando por tratar do carrinho que serviu de suporte para a máquina, e garantindo o fornecimento de água fria e água quente bem como o escoamento de águas residuais.

### 3.2.2 Instrumentação da máquina

O objetivo desta workpackage foi desenvolver um sistema de observação recorrendo a sensores e a um microcontrolador, de forma a conseguir capturar o estado de cada componente interno durante todo o ciclo de lavagem. Este sistema utiliza uma placa *Arduino Uno*.

### 3.2.3 Caracterização do ciclo de lavagem

Nesta Workpackage foi feita a caracterização da máquina de lavar bem como de um ciclo de lavagem completo. A caracterização foi feita por inspeção dos componentes internos e análise dos dados adquiridos pelos sensores durante o funcionamento da máquina.

### 3.2.4 Projeto do controlador

O objetivo desta Workpackage foi o de projetar um controlador suficientemente flexível de modo a que este tenha a capacidade de se adaptar a mais do que um tipo de máquina de lavar. Esta é a principal Workpackage do projeto e por isso é objeto de uma descrição mais detalhada.

A tarefa inicial desta Workpackage foi a definição dos requisitos de hardware e software, seguindo-se o estudo dos componentes necessários para a montagem do protótipo do controlador e o desenvolvimento do software necessário para que o controlador consiga receber informação dos sensores, processá-la e controlar os atuadores.

Depois passa-se para a montagem em *breadboard* de um primeiro protótipo do controlador e a integração do software desenvolvido no protótipo do controlador por forma a poder realizar os ensaios preliminares.

### 3.2.5 Ensaios preliminares e validação de resultados

O objetivo desta Workpackage foi testar o protótipo desenvolvido e verificar se se cumpriam os requisitos previamente definidos. Os resultados obtidos foram comparados com os obtidos na Workpackage 3, e foram efetuados os necessários ajustes ao controlador por forma a serem cumpridos todos os requisitos definidos.

### 3.2.6 Protótipo industrial

Nesta Workpackage desenvolveu-se o protótipo industrial do controlador, composto por duas placas de circuito impresso. Uma placa para o controlador e painel de controlo da máquina e uma placa com as interfaces de potência. De seguida procedeu-se à montagem das placas e ao desenho do novo painel de controlo local da máquina. Fez-se a recablagem da máquina e o teste final das PCBs e do software desenvolvido.

### **3.2.7 Desenvolvimento de funcionalidades adicionais**

Esta Workpackage teve como objetivo o desenvolvimento de funcionalidades adicionais, semelhantes às que as máquinas topo de gama possuem. Estas funcionalidades são o cálculo do custo do ciclo de lavagem com base na medição dos consumos de água e de energia e a possibilidade de informar o utilizador sobre o estado da máquina durante o seu funcionamento.

Fez ainda parte desta Workpackage o desenvolvimento de software de manutenção que permite aceder e controlar individualmente cada componente interno da máquina.

### **3.2.8 Acesso remoto**

Nesta Workpackage desenvolveu-se a aplicação móvel que permite comandar a máquina remotamente. A aplicação tem a capacidade de enviar comandos para a máquina e receber informação sobre o estado da máquina. A máquina recebe e executa os comandos recebidos e envia informação sobre o seu estados sempre que solicitado.

Inicialmente foram definidos os comandos necessários para desenvolver o controlo remoto, e o protocolo de comunicação através do qual os comandos e a informação são enviados do dispositivo móvel para o controlador e vice-versa.

### **3.2.9 Gestão do projeto**

Nesta Workpackage, que decorreu ao longo de todo o projeto, fez-se a gestão do projeto, verificando a correta e atempada a execução de todas as tarefas propostas, com especial atenção aos prazos estabelecidos. Desta forma a gestão do projeto foi feita de forma contínua ao longo do projeto, prevenindo atrasos no seu desenvolvimento.

## **3.3 Diagramas de Gantt**

Este trabalho estendeu-se por dois semestres letivos. Apresenta-se nesta secção o diagrama de Gantt da primeira fase (2017) e o da segunda fase (2018) por forma a melhor evidenciar as tarefas executadas em cada semestre.

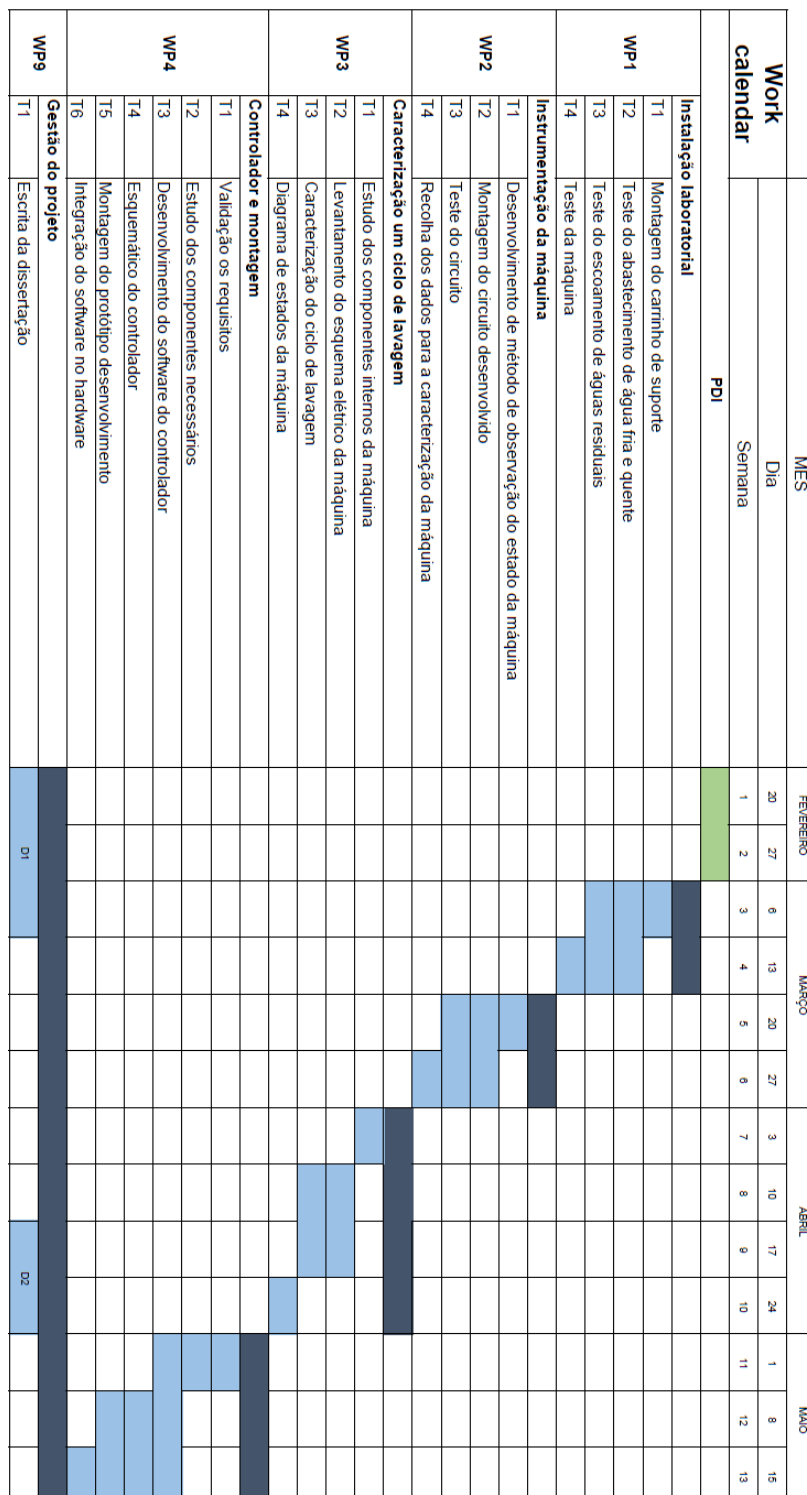


Figura 3.2: Diagrama de Gantt - 1ª fase - 2017

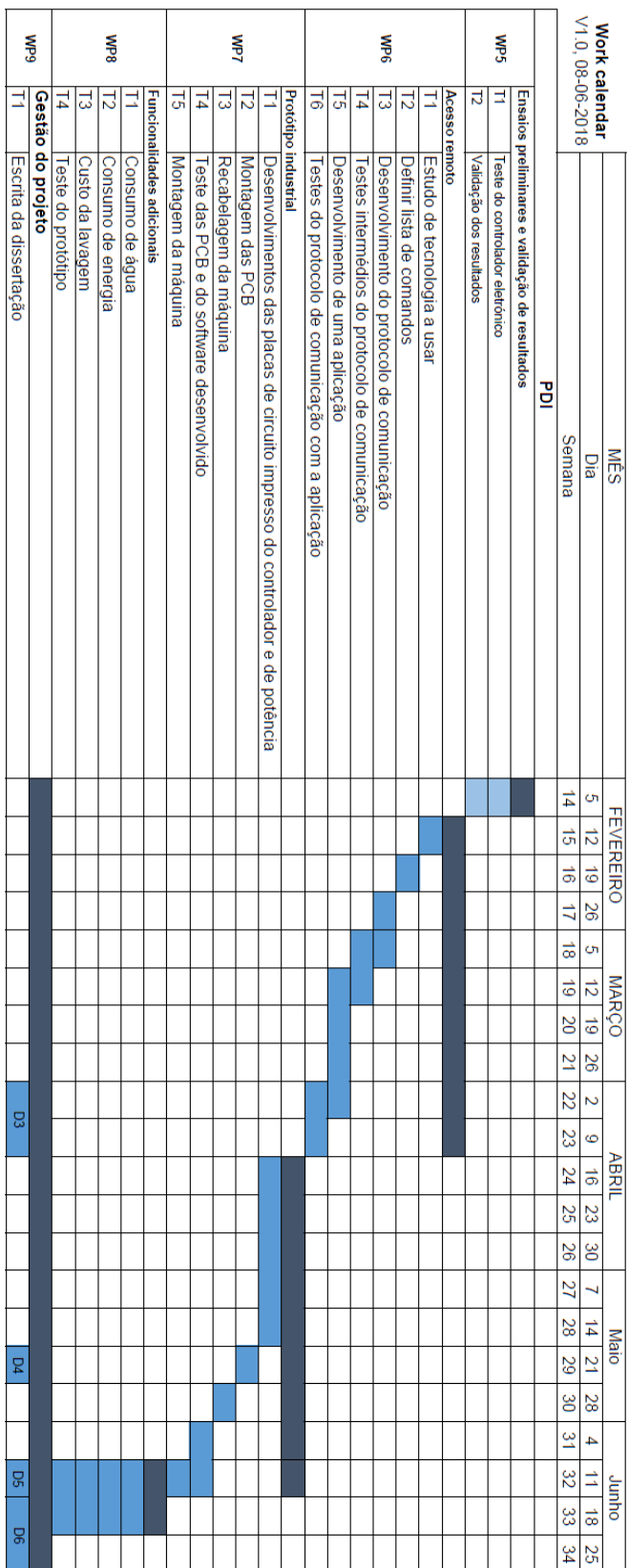


Figura 3.3: Diagrama de Gantt - 2ª fase - 2018

## Capítulo 4

# Caracterização da máquina e de um ciclo de lavagem

Neste capítulo é feita a caracterização da máquina, descrevendo detalhadamente os componentes internos e o ciclo de lavagem que esta implementa.

### Conteúdo

4.1	Funcionamento geral de uma máquina de lavar loiça . . . . .	13
4.2	Caracterização do ciclo de lavagem . . . . .	16
4.3	Componentes internos . . . . .	16

### 4.1 Funcionamento geral de uma máquina de lavar loiça

A lavagem é semelhante à lavagem manual, é necessário uma combinação de duas ações, a mecânica e a química. As diferenças são a temperatura muito mais elevada e substituição da escovagem pela circulação forçada da água.

A elevada temperatura atingida assegura uma lavagem mais eficaz e uma melhor desinfecção da loiça. O uso da máquina de lavar loiça é vantajoso no que toca ao consumo de água. Em média uma pessoa usa 60 litros de água (se usar a água a 45 °C) para lavar a loiça à mão enquanto que uma máquina de lavar consome em média metade desse valor e, dependendo do modelo da máquina o consumo pode descer até aos 6,5 litros [3]. Em traços gerais, uma máquina simples executa as seguintes operações:

1. Adiciona água limpa
2. Aquece a água até temperatura desejada (80°C no modelo estudado)
3. Abre o dispensador de detergente e abrillantador no momento adequado
4. Bombeia a água limpa durante um determinado tempo

5. Drena a água suja

6. Aquece o ar no interior da máquina para ajudar a secar a loiça

Algumas destas operações são feitas mais do que uma vez durante um ciclo de lavagem, nesta máquina as operações 1 até à 5 repetem-se até 3 vezes.

Máquinas sofisticadas possuem outras funções, tais como:

- Opção de regular a duração de cada ciclo de lavagem
- Sensor de temperatura, prevenindo que a máquina sobreaqueça
- Sensor de nível da água para ativar a bomba de drenagem e evitar que a máquina transborde
- Sensor de turbidez da água de lavagem para ser possível controlar a duração do ciclo de lavagem e a água consumida em função da sujidade da loiça

#### **4.1.1 Ciclo de lavagem de uma máquina simples**

Um ciclo de lavagem da loiça começa por encher a máquina de água fria através de uma mangueira que liga a máquina ao abastecimento de água da casa. Depois de ter quantidade suficiente de água no seu interior, a resistência de aquecimento que está normalmente situada no fundo da máquina é ativada e por contacto aquece gradualmente a água durante o ciclo de lavagem.

De seguida é ativada a bomba de circulação que leva a água quente até duas pás rotativas (uma debaixo da prateleira inferior outra debaixo da prateleira superior). Quando a água pressurizada entra nas pás estas giram como se fossem aspersores de jardim expelindo a água através de orifícios situados na sua superfície. Os jatos de água efetuam a lavagem mecânica da loiça. Depois da água bater na superfície da loiça, escorre para o fundo da máquina onde é aquecida e bombeada novamente. Este ciclo é repetido até ser acionada a bomba de drenagem para dar início à última fase do ciclo de lavagem.

Por fim, como a temperatura no interior da máquina no fim do ciclo de lavagem é elevada a água à superfície da loiça evapora e desta forma a loiça fica seca naturalmente. Em algumas máquinas é usada a resistência de aquecimento para ajudar à secagem da loiça.

#### **4.1.2 Necessidade do uso de detergente e sal**

Para que uma máquina de lavar loiça consiga fazer uma lavagem eficiente é necessário o uso de detergente para ajudar a separar a gordura e restos de comida da loiça.

O detergente é uma mistura de compostos para estimular a separação dos diferentes restos de comida da loiça e talheres.



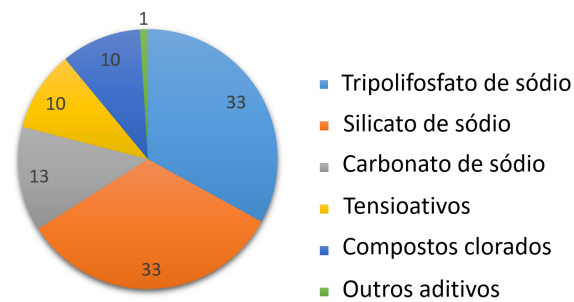


Figura 4.1: Composição típica de um detergente

A composição típica de um detergente está apresentada na figura 4.1

- Tensioativos - para diminuir a tensão superficial da água de modo a mais facilmente remover e emulsionar restos gordurosos, como a manteiga e outros óleos de cozinha. Alguns agentes tensioativos são nocivos para o ambiente por promoverem a eutrofização do meio aquático existindo por isso legislação que restringe as suas concentrações nos detergentes
- Tripolifosfato de sódio - Reage com os iões metálicos da água como os iões cálcio e iões magnésio, formando uma solução e deste modo impedindo que estes possam reagir com outros componentes. Também ajuda a manter o nível de alcalinidade desejado
- Silicato de sódio - previne a corrosão de metais, protegendo os componentes da máquina e as porcelanas com dourados
- Carbonato de sódio - ajuda a remover componentes gordurosos
- Compostos clorados - ajudam a dissolver componentes proteicos dos alimentos como o ovo e o leite e auxilia na remoção de manchas de café e chá
- Outros - enzimas e fragrâncias

Os fabricantes de detergentes seguem leis que limitam o uso destes compostos, por forma a evitar poluir em excesso as águas. Porém é responsabilidade do utilizador fazer um uso inteligente destes detergentes. A quantidade a usar deve ser suficiente de forma a proporcionar uma boa lavagem. [4, 5]

#### 4.1.3 Abrilhantador

O abrilhantador é um químico que reduz a tensão superficial da água, normalmente é constituído por surfactantes. Este químico ao reduzir a tensão superficial da água evita que se formem gotículas de água na superfície da loiça, facilitando assim a secagem da

loíça e evitando manchas residuais. Este agente não dá brilho á loíça, simplesmente evita que os minerais dissolvidos na água fiquem depositados nas superfície da loíça quando a água evapora tornando-a menos brilhante.

## 4.2 Caracterização do ciclo de lavagem

A caracterização do ciclo de lavagem da máquina, tem como objetivo a futura reprodução do ciclo atual, com o novo controlador e passa por registar temporalmente todos os acontecimentos durante o processo. Estes acontecimentos são a ativação das válvulas de entrada de água, a ativação do motor, a abertura do compartimento do detergente, ativação do motor de escoamento de águas, bem como as variáveis de processo (temperatura, consumo de água e energia).

## 4.3 Componentes internos

Uma máquina de lavar loiça (figura 4.2) é um eletrodoméstico relativamente simples. Uma bomba de circulação faz circular a água em circuito fechado durante o ciclo de lavagem. A loiça é lavada pela ação química do detergente e pela ação mecânica da água circulante. A elevação da temperatura aumenta a eficácia da lavagem.

A máquina utilizada neste estudo foi uma máquina da marca Balay, modelo V4521. Trata-se de uma máquina de gama média, já relativamente antiga, equipada com um controlador eletromecânico e possibilitando a escolha entre 1 de 4 programas de lavagem distintos:

- P1: Pré-lavagem
- P2: Lavagem reduzida (meia carga)
- P3: Lavagem normal
- P4: Lavagem intensiva

Os componentes internos da máquina estão representados na tabela 4.1 e se descrevem de seguida.

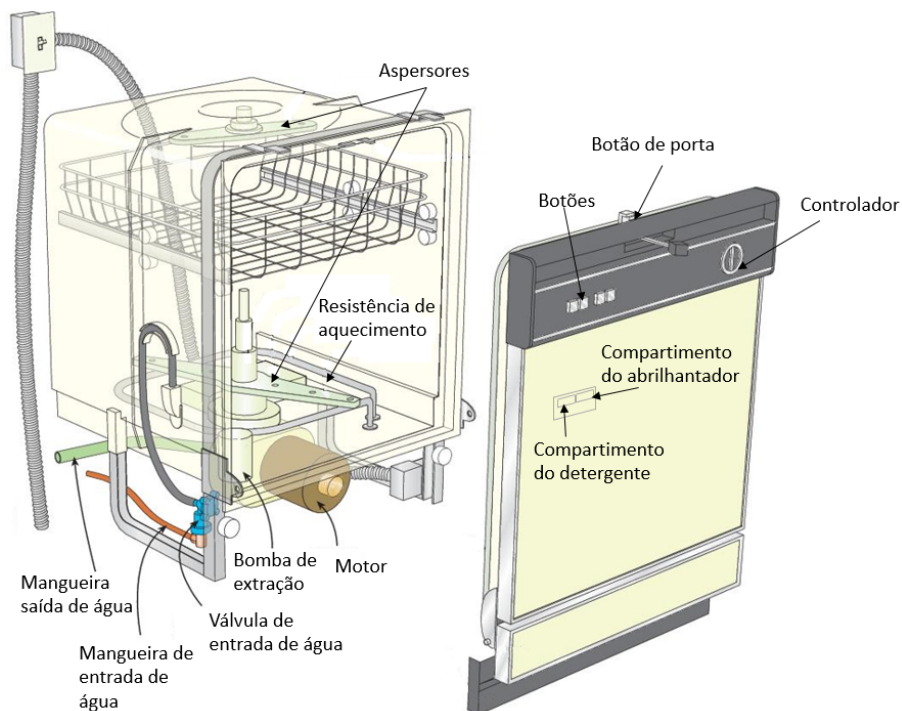


Figura 4.2: Modelo de uma máquina de lavar louça simples [6]

Tabela 4.1: Componentes atuadores da máquina Balay V4521

Componente	Características
Bomba de circulação	Motor de indução monofásico
Bomba de drenagem	Motor de polos sombreados Potência: 34 W
Resistência de aquecimento	Potência: 1950 W
Acondicionamento de água	Resina de permuta iônica
Dispensador de Detergente e líquido abrillantador	gaveta c/ acionamento eletro-mecânico
Controlador	
Válvula de entrada	Eletro-válvula simples

#### 4.3.1 Bomba de circulação

É acionada por um motor de indução monofásico. Este tipo de motores é usado em aplicações domésticas onde o abastecimento de energia com três fases não é possível ou conveniente. A potência deste tipo de motores pode atingir 2.2 kW, sendo a maioria inferior a 500 W. [7, p. 227]

### 4.3.2 Bomba de extração

Acionada por um pequeno motor de polos sombreados. Este tipo de motor é extremamente simples e robusto e é normalmente aplicado em situações de baixa potência e trabalhos unidirecionais, que é precisamente o que se pretende para a bomba de extração de água da lavagem [7, p. 231-232].

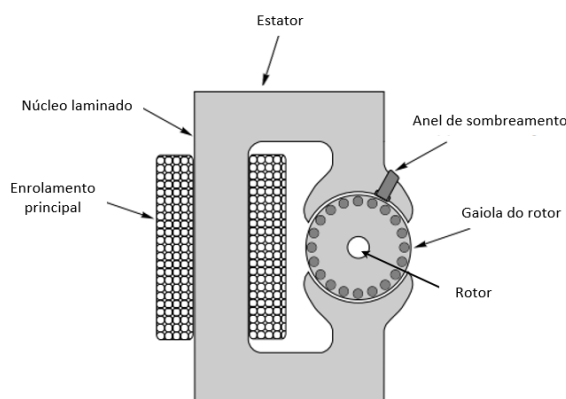


Figura 4.3: Motor de indução de polos sombreados [7, p. 231]

### 4.3.3 Resistência de aquecimento

É uma resistência metálica de elevada potência (neste caso  $\approx 1,85KW$ ) que tem como função aquecer a água e todo o ambiente em que a loiça está inserida.

### 4.3.4 Circuito de acondicionamento da água de lavagem

Trata de fazer o acondicionamento da água reduzindo a sua dureza. A dureza da água é causada pela presença de iões de cálcio e magnésio dissolvidos na água. Se estes iões forem removidos a água torna-se mais macia, evitando a acumulação de calcário no interior da máquina e na superfície da loiça e aumentando a eficácia de lavagem.

Neste componente existe um depósito de uma resina de permuta iónica, na forma de pequenas esferas com raio de 0,25 a 0,5 mm, fabricadas a partir de um substrato de polímero orgânico. As esferas são normalmente porosas para aumentar a superfície de contacto com a água. Existem muitos tipos de resinas de permuta iónica, sendo a maioria feita de sulfonato de poliestireno [8, 9]. Nesta resina dá-se a troca de iões de cálcio e magnésio ( $Ca^{++}$  e  $Mg^{++}$ ) presentes na água, por iões de sódio ( $Na^{+}$ ) que não interferem no processo de lavagem nem alteram a eficiência de lavagem. Para que isto aconteça a água limpa passa pela resina antes de entrar na cuba da máquina. Quando a água flui através da resina efetua-se a troca iónica, iões de sódio são libertados e os iões de magnésio e cálcio absorvidos pela resina formam um composto que não afeta a eficiência de lavagem da máquina. De forma a recuperar a capacidade da resina de permutar iões é necessário

repor a concentração de íons de sódio na superfície da resina, para isso existe um outro depósito com cloreto de sódio ligado ao primeiro, que por ativação de uma válvula simples abre a ligação entre estes dois depósitos.[8, 9]

#### 4.3.5 Dispensador de detergente e líquido brilhantador

Trata-se de uma gaveta com um acionamento eletro-mecânico que liberta o detergente no momento desejado. Neste modelo de máquina, o acionamento da gaveta também faz bombear uma pequena quantidade de líquido brilhantador para a cuba de lavagem.

#### 4.3.6 Termostatos

Esta máquina possui dois termostatos. O termostato de funcionamento desliga a resistência assim que a água atinja a temperatura de 78 °C. Caso haja uma avaria deste componente existe um segundo termostato, de segurança, calibrado para 85 °C que desligará a máquina, impedindo que esta sobreaqueça.

#### 4.3.7 Ciclo de lavagem de máquina Balay V4521

Cada marca faz pequenos ajustes a este ciclo de lavagem genérico por isso foi necessário fazer uma caracterização detalhada do ciclo de lavagem deste modelo. Podendo assim definir como é que são os quatro programas de lavagem que a máquina dispõe. O programa P1 representa a pré-lavagem, o programa P2 efetua uma lavagem com meia carga, o programa P3 efetua uma lavagem de carga completa e o programa P4 efetua uma lavagem de carga completa com loiça muito suja.

Depois de recolhidos e tratados todos os dados podem-se caracterizar os programas disponibilizados pela máquina. A figura 4.4 representa a pré lavagem ou programa P1.

Este programa dura aproximadamente 7 minutos e meio. É de notar que a unidade do eixo horizontal é o segundo. Como se pode observar, na pré-lavagem há apenas circulação de água com pressão por forma a tirar os restos de comida maiores, no final é feita a drenagem da água suja. A água limpa antes de entrar na cuba da máquina passa para um depósito, de forma a que o controlador eletromecânico tenha controlo da água consumida. Para isto, quando um depósito de água está cheio um sinal é enviado ao controlador. Neste caso são necessários três depósitos para a pré lavagem.

O programa P4 é o ciclo de lavagem mais longo que a máquina disponibiliza e é indicado para uma lavagem com a carga completa e loiça muito suja. O programa P2 é semelhante mas para loiça pouco suja e o programa P3 é o que o fabricante recomenda para meia carga. A figura 4.5 apresenta os resultados da monitorização dos programas P4, P3 e P2.

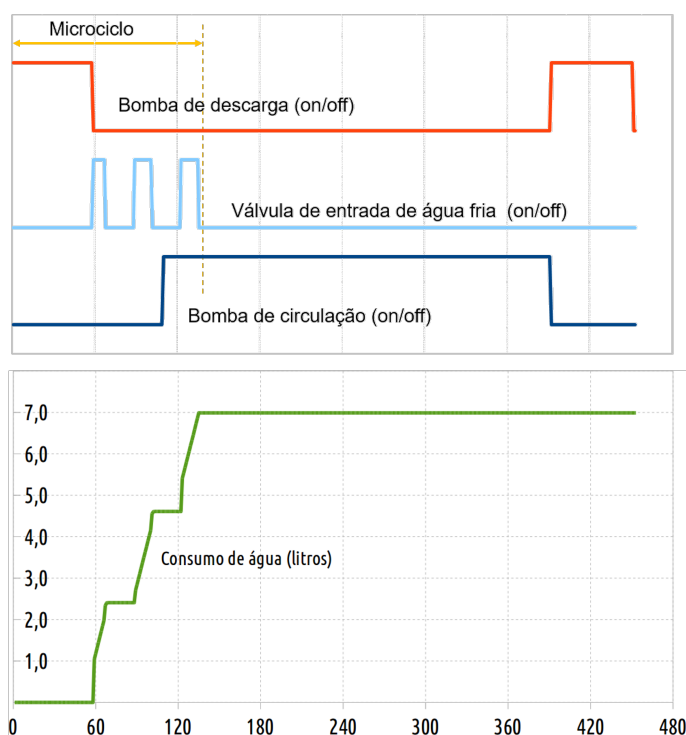


Figura 4.4: Pré lavagem levantado através do Datalogger

Neste gráfico a unidade do eixo horizontal é o minuto. O eixo vertical esquerdo indica a temperatura em  $^{\circ}\text{C}$ ) e o eixo vertical direito indica o consumo de água em litros.

A diferença entre os programas P4, P3 e P2 é apenas o número de pré-lavagens com água fria que são feitas antes da lavagem principal com água quente. Por outras palavras, os programas P3 e P2 são em tudo semelhantes, apenas começam numa fase mais avançada do programa P4.

Na figura 4.5 é possível observar os diferentes sinais ao longo do ciclo de lavagem. É de notar que o sinal a verde (Detergente/abrillantador) muda de significado durante o ciclo de lavagem, uma vez que o sistema de libertação do detergente está ligado ao sistema de libertação de líquido abrillantador. Na primeira ativação desse sinal é libertado o detergente (e também líquido abrillantador); nas restantes ativações a gaveta do detergente já está aberta por isso essas ativações resultam apenas na libertação de abrillantador.

#### 4.3.8 Diagrama de estados da máquina

Os resultados recolhidos permitiram construir o diagrama de estados da máquina. De forma a facilitar a leitura, o diagrama de estados foi dividido num sub diagrama que representa um microciclo que se repete várias vezes ao longo de qualquer ciclo de lavagem. Na figura 4.6 está representado esse microciclo, que corresponde à zona assinalada na figura 4.4.

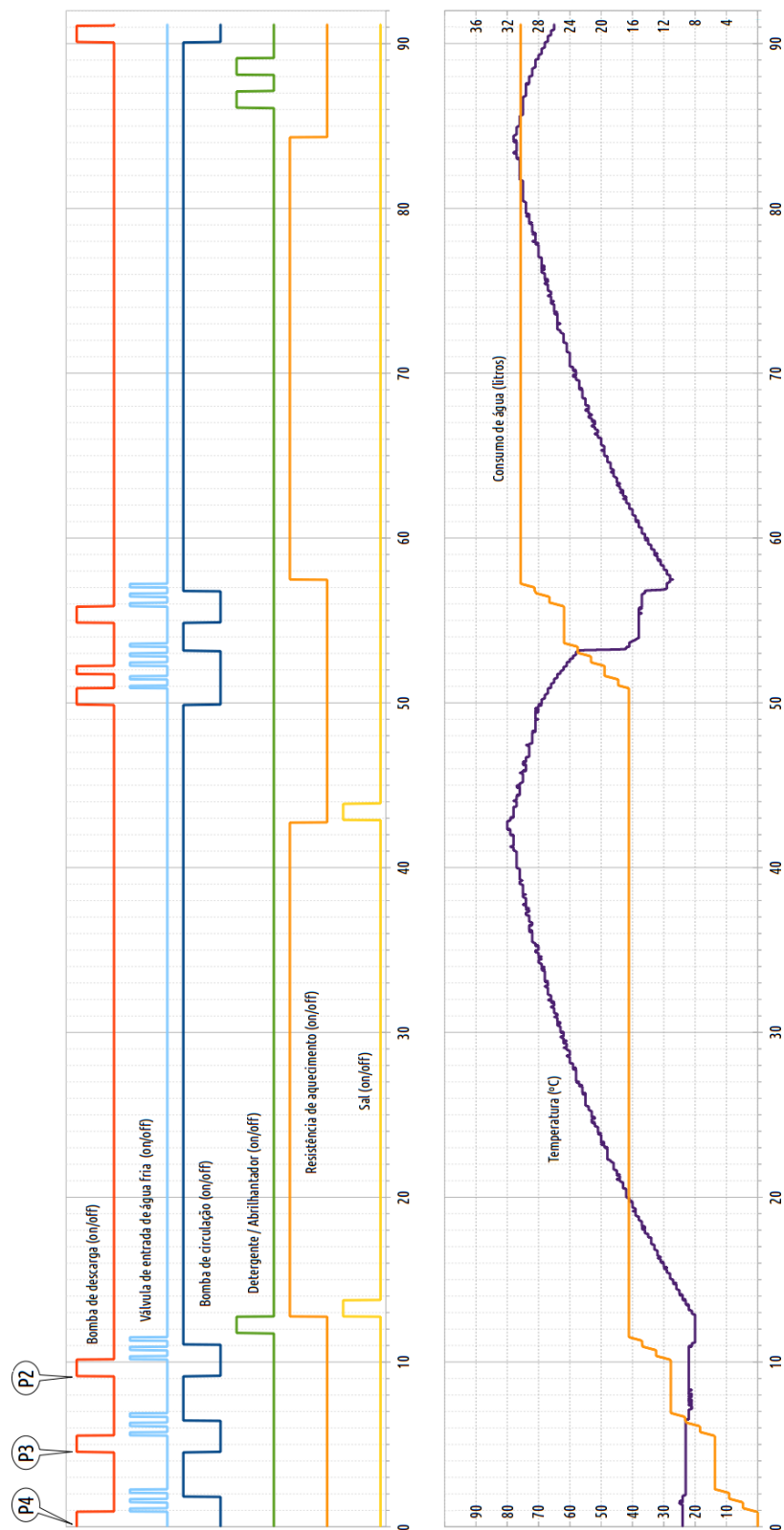


Figura 4.5: Programa de lavagem levantado através do Datalogger

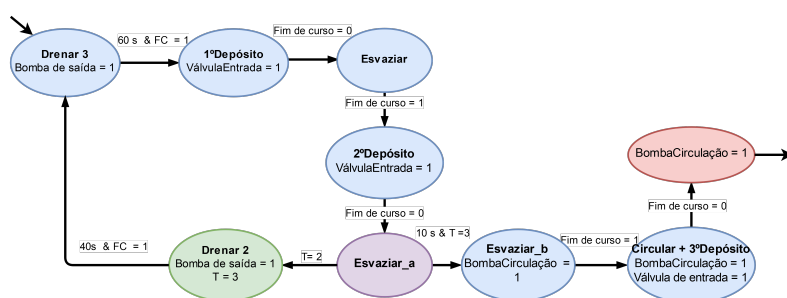


Figura 4.6: Diagrama de estados relativo ao microciclo

A figura 4.7 representa o diagrama de estados completo que representa o funcionamento geral da máquina de lavar.

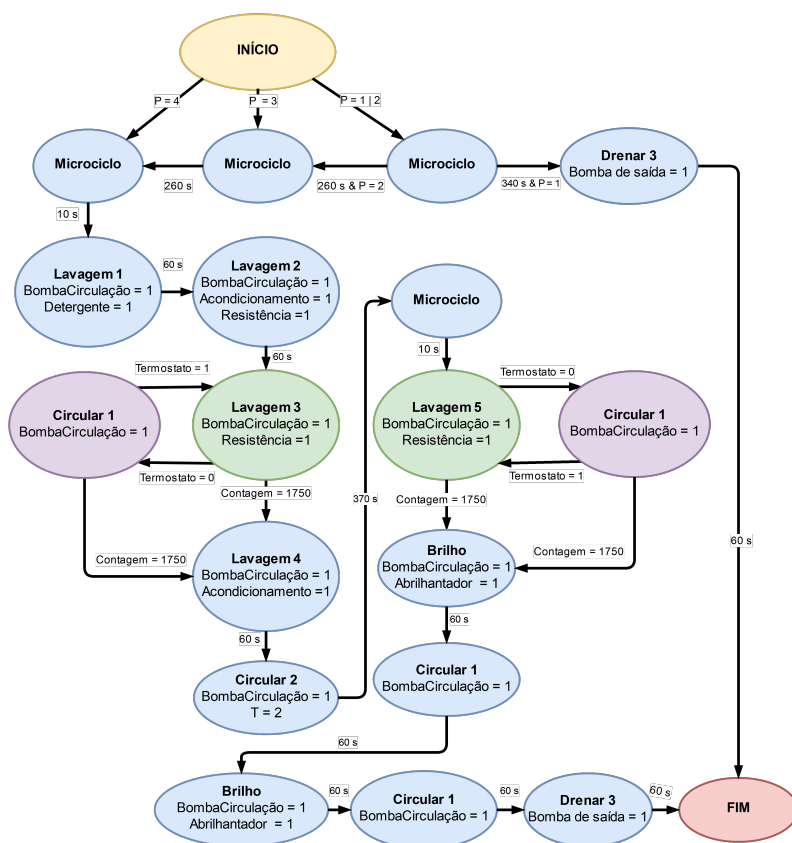


Figura 4.7: Diagrama de estados da máquina



## Capítulo 5

# Instalação e instrumentação da máquina

Neste capítulo está descrito como foi preparado o local de trabalho e o método de instrumentação da máquina desenvolvido para a caracterização do ciclo de lavagem.

### Conteúdo

---

5.1	Preparação do laboratório e instalação da máquina . . . . .	23
5.2	Instrumentação da máquina . . . . .	24
5.3	Calibração do caudalímetro . . . . .	26

---

### 5.1 Preparação do laboratório e instalação da máquina

Nesta fase preparou-se todo o local de trabalho, desde o carrinho em que a máquina se encontra até à bancada com o material necessário para realizar o estudo. O local de trabalho foi o laboratório I002, e aí instalou-se a máquina de lavar, junto da rede de abastecimento e escoamento de águas, revistas expressamente para este fim pelos serviços técnicos e de manutenção da FEUP. Procurou-se que o interior da máquina fosse de fácil acesso e para isso a porta foi substituída por um painel transparente em policarbonato.

Verificou-se o correto funcionamento da máquina com o controlador original e procedeu-se a uma breve inspeção dos componentes internos na máquina e ao levantamento do esquema elétrico completo, com o objetivo de facilitar a implementação do novo controlador eletrónico. O esquema elétrico da máquina está apresentado na figura 5.1. Posteriormente foi efetuado um levantamento dos materiais necessários para o desenvolvimento do projeto, nomeadamente os sensores e conectores.

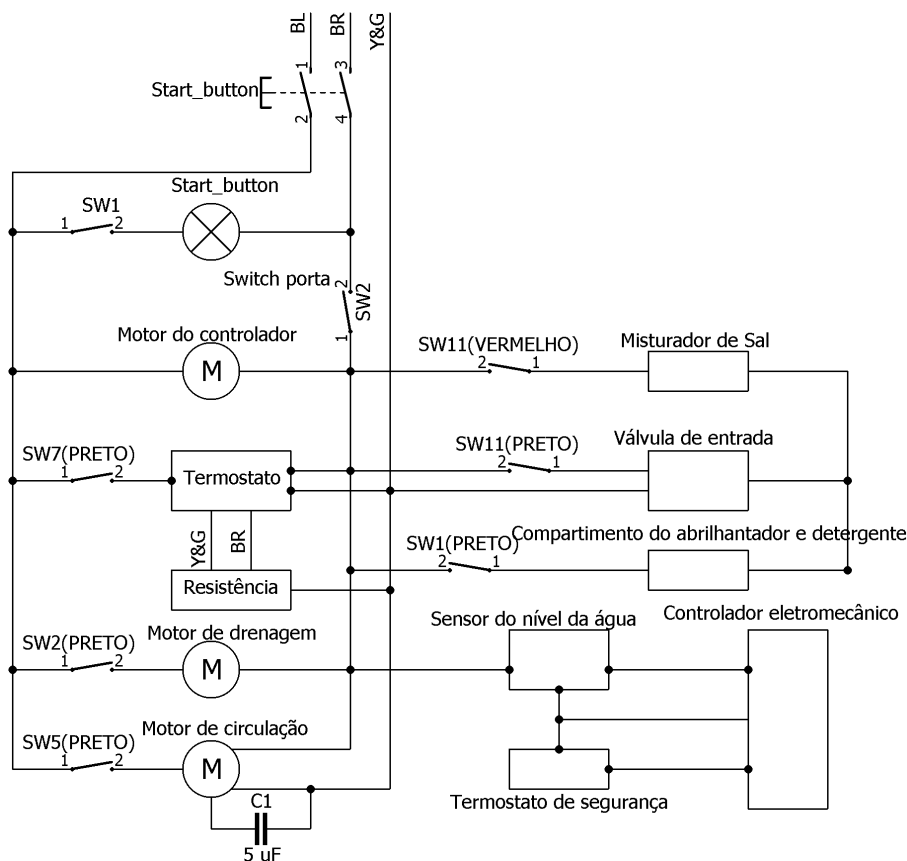


Figura 5.1: Esquema elétrico original da máquina

## 5.2 Instrumentação da máquina

Para registrar o estado dos componentes durante um ciclo de lavagem, projetou-se um circuito *Datalogger* com entradas isoladas opticamente, uma vez que a maioria dos componentes internos da máquina funciona a 230 Vac. O consumo de água e a temperatura foram também adquiridos recorrendo a um caudalímetro e a um sensor de temperatura, respetivamente.

O *Datalogger* faz uso de um optocoupler para fazer o drive dos 230 Vac que alimentam todos os componentes internos da máquina dos 5 V que alimentam os componentes eletrónicos. A figura 5.2 representa o esquema do circuito de isolamento de uma entrada.

Este circuito foi replicado 8 vezes e está ligado a entradas digitais de um microcontrolador ATMEGA328p.

Para a captação da temperatura foi usado o sensor de temperatura LM35DT ligado a uma entrada analógica. Para medir o consumo de água colocou-se um caudalímetro na mangueira de entrada de água, ligado a uma entrada digital não isolada.

A figura 5.3 apresenta o esquema elétrico completo do circuito desenvolvido para recolher toda a informação acerca do ciclo de lavagem.

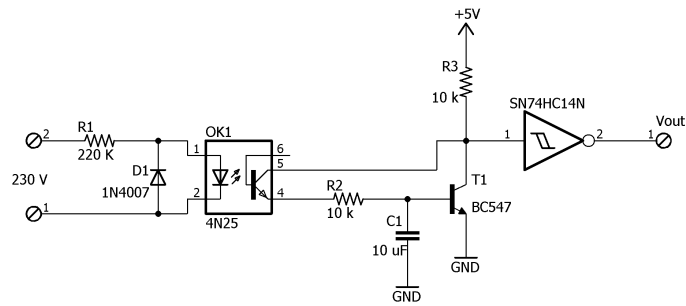


Figura 5.2: Esquema elétrico do circuito de isolamento de uma entrada

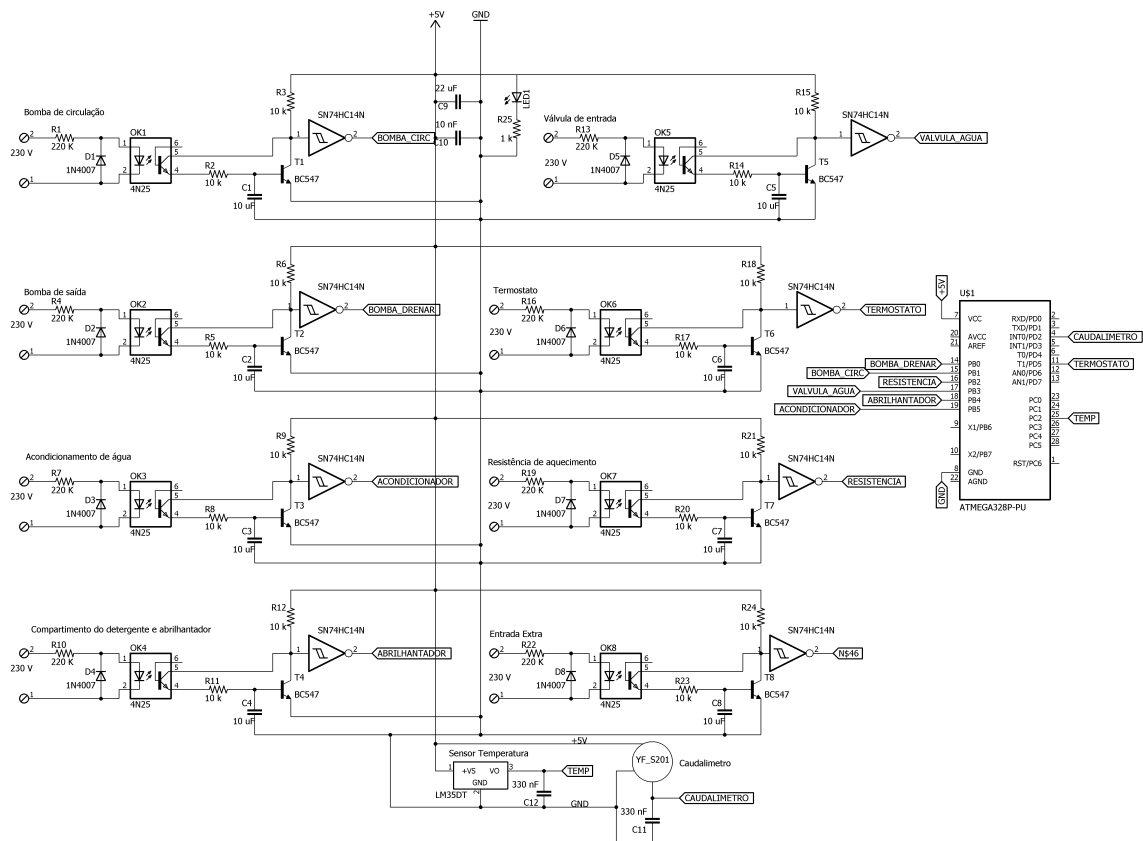


Figura 5.3: Esquemático do circuito Datalogger

Tendo montado todo o hardware necessário, foi desenvolvido o software para capturar o estado de cada componente, a temperatura da máquina, o consumo de água e posteriormente enviar esses valores via porta série para que pudessem ser guardados num ficheiro.

### 5.2.1 Software para recolha de dados

O software desenvolvido teve como objetivo caracterizar o ciclo de lavagem completo. Para isso, a cada segundo, foi registado o estado de cada componente, a temperatura e o consumo de água e os dados enviados para um computador e armazenados num ficheiro

para posterior análise e processamento. A contabilização do consumo de água foi feita em contínuo pois o caudalímetro estava ligado a uma interrupção externa independente. A base de tempo de aquisição de 1 segundo e foi implementada pela interrupção periódica de um *timer*. Essa interrupção desencadeava a leitura de entrada das entradas digitais e da entrada analógica e o envio dessa informação, bem como o consumo de água, pela porta série.

Do lado do computador foi necessário captar a informação recebida e guardá-la num ficheiro de texto. Neste caso foi utilizada a aplicação *CoolTerm* por facilidade de utilização. O software de aquisição constitui o anexo [A](#) deste documento.

### 5.3 Calibração do caudalímetro

Foi necessário proceder à calibração do caudalímetro pois a indicação dada pelo fabricante, após alguns ensaios, levantou dúvidas. Para isso fez-se passar pelo caudalímetro um volume de água conhecido e registou-se o número de impulsos gerados. Com um micro-controlador e um contador de eventos (Data Precision, modelo 5740) contaram-se os impulsos e com uma balança de precisão (C&K, modelo 6000) mediu-se a massa de água. Os resultados das medições estão apresentados na tabela [5.1](#):

Tabela 5.1: Dados para calibração do caudalímetro

Massa de água (g)	Impulsos (Contador)	Impulsos (Arduino)
4585.5	1808	1813
4696.4	1903	1854
4514.0	1667	1773
4656.9	2111	1846

Considerando que a densidade da água a 20°C) é  $998 \text{ kg/m}^3$ , obtiveram-se os resultados descritos na tabela [5.2](#) que, após determinação da incerteza, permitem garantir uma melhoria de 10 vezes na precisão da medida, passando dos 10% garantidos pelo fabricante para 1%. Isto é, de  $450 \pm 45$  para  $394 \pm 3$  impulsos por litro.

Tabela 5.2: Volume de água (litros) e impulsos por litro

Volume de água	Impulsos (Arduino)	Impulsos (Contador)
4.5947	394.6	393.5
4.7058	392.0	393.6
4.5230	395.6	407.7
4.6662	394.0	407.8
Média	394.0	400.1
Desvio padrão	1.52	
Incerteza	3	

## Capítulo 6

# O novo controlador da máquina

Neste capítulo descreve-se o *hardware* do controlador eletrónico desenvolvido para substituir o controlador eletromecânico original e apresenta-se a nova cablagem da máquina, substancialmente mais simples do que a original.

### Conteúdo

6.1	Cablagem e diagrama de blocos . . . . .	27
6.2	Requisitos . . . . .	28
6.3	Módulo de controlo . . . . .	29
6.4	Módulo de potência . . . . .	30
6.5	Alterações posteriores . . . . .	33

### 6.1 Cablagem e diagrama de blocos

A figura 6.1 representa a nova cablagem e diagrama de blocos da máquina. O módulo de controlo trata do controlo do sistema e o módulo de potência agrupa todas as interfaces de potência necessárias.

Um interruptor geral, já existente na cablagem original, faz a interrupção da alimentação de todo o circuito. Uma fonte de alimentação comutada fornece a alimentação aos módulos de controlo e de potência. Para esta fonte optou-se por utilizar um alimentador de telemóvel por ser a solução com melhor relação custo/benefício. O módulo de controlo recebe dados do sensor de temperatura, do caudalímetro e do estado do interruptor da porta e fornece a alimentação para o módulo de potência bem como os sinais necessários à comunicação entre os módulos, neste caso comunicação SPI. O interruptor da porta, quando a mesma está aberta, interrompe a alimentação de todos os componentes eletromecânicos da máquina. Esta medida introduz um nível de segurança adicional à paragem imposta também pelo *software* de controlo.

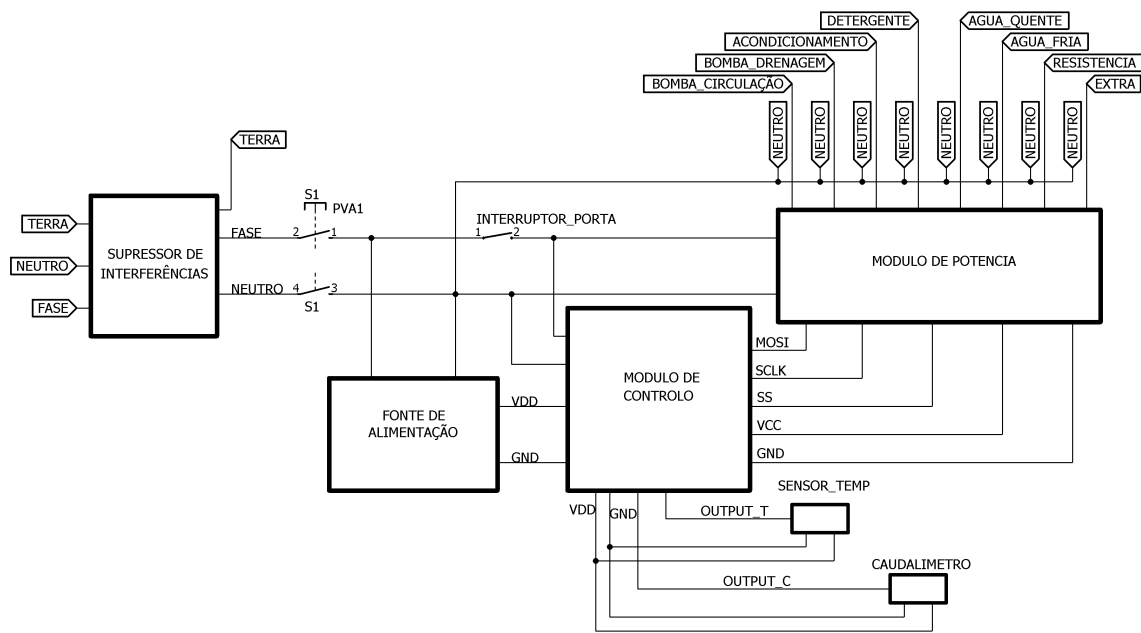


Figura 6.1: Cablagem e diagrama de blocos da máquina

## 6.2 Requisitos

A caracterização da máquina, dos ciclos de lavagem, a definição do novo modo de funcionamento e novas funcionalidades permitiram definir os requisitos de *hardware* para o controlador a desenvolver. Assim, para garantir a implementação dos ciclos de lavagem originais são necessárias:

- Uma entrada digital isolada (interruptor da porta)
- Uma entrada digital não isolada (fim de curso de enchimento)
- Seis saídas digitais isoladas (bomba de drenagem, válvula de entrada de água, bomba de circulação, resistência de aquecimento, dispensador de detergente, regenerador da resina)

Por outro lado, para assegurar a nova interface local com o utilizador são necessárias:

- Cinco saídas digitais não isoladas (LEDs)
- Três entradas digitais não isoladas (codificador incremental com botão de pressão)

Finalmente, para assegurar algumas das funcionalidades adicionais de que se pretende dotar o novo controlador (medição da temperatura, medição do consumo de água, possibilidade de ligação a uma rede de água quente e possibilidade de comando remoto) são necessárias:

- Uma entrada analógica (sensor de temperatura)
- Uma entrada digital não isolada (caudalímetro)

- Duas saídas digitais isoladas (válvula de entrada de água quente e saída extra)
- Uma porta série bidirecional (para comunicação remota)

O isolamento das entradas/saídas passa por fazer uso de um drive de isolamento do 230 Vac que alimentam todos os componentes da máquina dos 5 V que alimentam a maior parte dos componentes eletrónicos.

Resumindo, do ponto de vista do *hardware*, o novo controlador deverá possuir treze saídas digitais, oito das quais isoladas; cinco entradas digitais, uma das quais isolada; uma entrada analógica e uma porta série bidirecional.

### 6.3 Módulo de controlo

O objetivo principal deste módulo é tratar de todo o controlo, desde o tratamento dos dados recebidos do sensor de temperatura, do caudalímetro, da porta série, do botão de comando e do circuito de deteção da abertura da porta. O controlo é feito por comandos enviados via SPI para o módulo de potência. O módulo de controlo assegura também a interface local com o utilizador através de um botão rotativo (*rotary encoder*) e de pressão, 5 LEDs que informam o utilizador do estado da máquina. O significado dos LEDS é descrito capítulo 7. Este módulo faz uso de uma plataforma de prototipagem rápida *Arduino Nano*, baseada num microcontrolador ATMEGA328p. De forma a detetar a abertura da porta, foi necessário adicionar uma entrada digital isolada, para deteção do estado da porta (aberta/fechada) uma vez que esta quando aberta interrompe a alimentação(230 Vac) a todos os componentes internos da máquina. O caudalímetro, o botão de pressão e botão rotativo foram ligados a entradas digitais não isoladas e deverão poder funcionar por interrupção, para maior facilidade de desenvolvimento do *software*. As ligações ao microcontrolador estão listadas na tabela 6.1.

A figura 6.2 representa o esquema elétrico completo do módulo de controlo.

Na ligação do modulo Bluetooth foi necessário incluir um adaptador de tensão simples (divisor de tensão) pois o modulo Bluetooth funciona a 3.3 V. O circuito de deteção da abertura da porta é semelhante ao utilizado na aquisição de dados do Datalogger, à exceção do schmitt-trigger na saída do sinal que foi substituído por um condensador de 10  $\mu$ F.

Na figura 6.3 podem ser observadas as duas faces da placa de circuito impresso na sua versão final.

No desenho do circuito impresso começou-se por colocar os componentes de forma a minimizar a área, colocando os LEDS e o botão alinhados horizontalmente. Porém, por razões estéticas optou-se por redesenhar a placa por forma a que o botão (*rotary encoder*) e os LEDS ficassem mais distantes entre si.

Tabela 6.1: Ligações ao Arduino Nano

Porta	Tipo	Descrição
A0	Entrada analógica	Recebe a tensão do sensor de temperatura
A1	Interrupção externa	Recebe impulsos do caudalímetro
A2	Saída digital	Ligada ao LED1 do painel da máquina
5V		Alimentação da placa
GND		Massa
Tx	Saída digital	Saída da porta série
Rx	Entrada digital	Entrada da porta série
D2	Interrupção externa	Ligada ao rotary encoder
D3	Interrupção externa	Ligada ao botão de pressão
D4	Entrada digital	Ligada ao rotary encoder
D5	Entrada digital	Ligada ao circuito de deteção da abertura da porta
D6	Saída digital	Ligada ao LED2 do painel
D7	Saída digital	Ligada ao LED3 do painel
D8	Saída digital	Ligada ao LED4 do painel
D9	Saída digital	Ligada ao LED5 do painel
D10	Saída digital	Slave select do shift-register da placa de potência
D11	Saída digital	MOSI do shift-register da placa de potência
D12	Saída digital	Ligada ao fim de curso de enchimento
D13	Saída digital	SCLK do shift-register da placa de potência

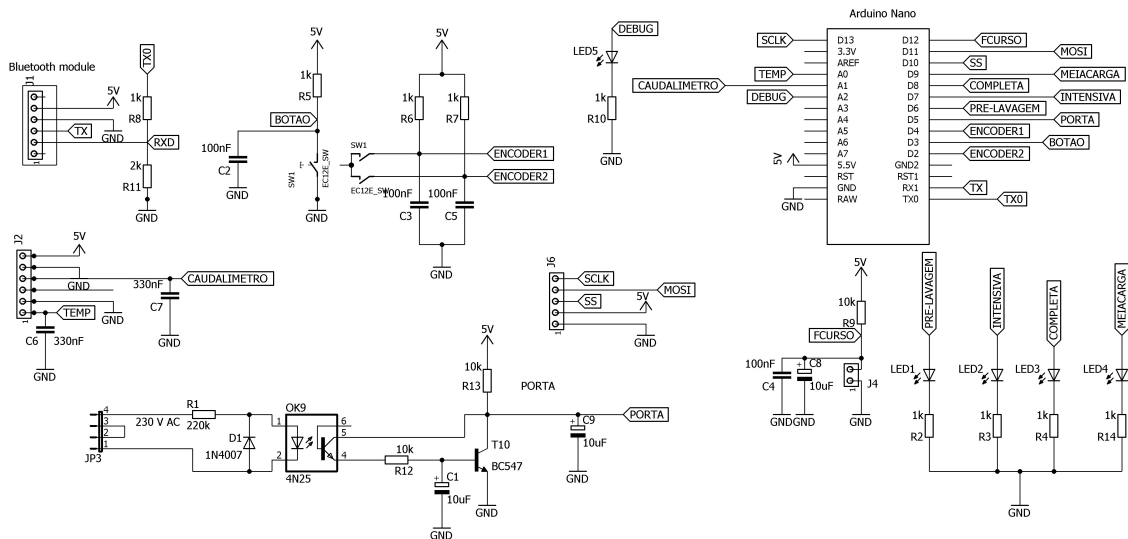


Figura 6.2: Esquemático da placa de controlo

## 6.4 Módulo de potência

O objetivo principal deste módulo é fazer o drive de tensão da rede elétrica de 230 V que alimenta todos os componentes da máquina dos 5 V utilizados como alimentação dos componentes eletrónicos do controlador.



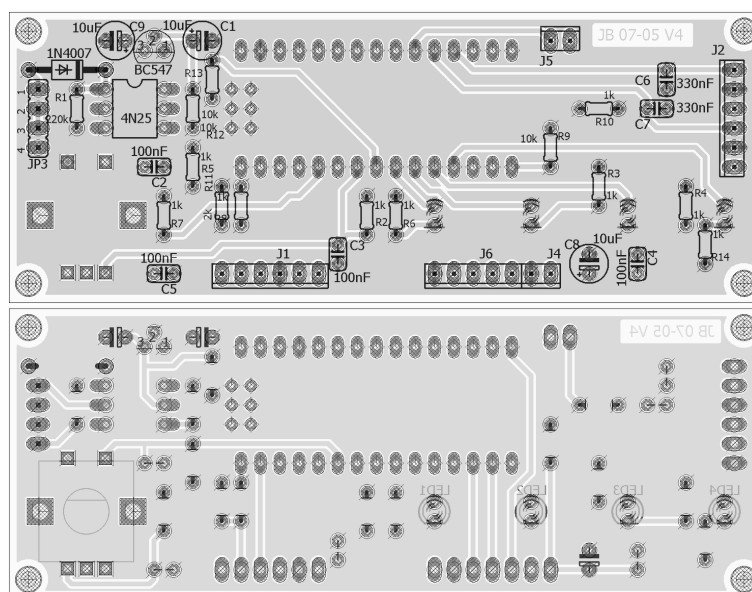


Figura 6.3: Placa de circuito impresso do módulo de controlo, Top/Bottom

As saídas digitais isoladas devem permitir controlar cargas funcionando a 230 Vac, algumas de potência elevada (por exemplo, a resistência de aquecimento da água tem uma potência nominal de 1.85 kW).

Um *shift-register* recebe oito bits de comando da placa de controlo (comunicação SPI). Cada um desses oito bits representa o estado de uma saída digital isolada que controla um componente da máquina. Estar a 1 significa ligar a respetiva saída, estar a 0 significa desligá-la. As saídas dispõem de triacs isolados opticamente tal como representa a figura 6.4.

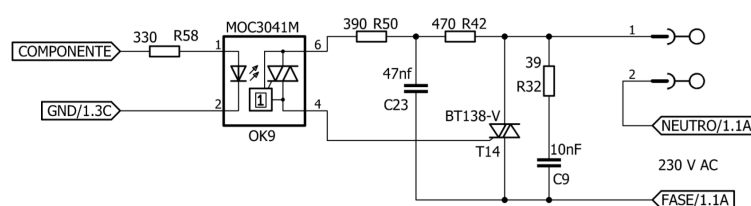


Figura 6.4: Circuito de drive dos 5 V para os 230 Vac

Se a entrada tiver o valor lógico 1 o LED do optotriac *MOC3041* emite luz fazendo com que o triac conduza que por sua vez faz conduzir o triac de potência *BT138* ativando o componente da máquina correspondente.

Este circuito foi replicado 8 vezes, para se poder controlar 8 dispositivos eletromecânicos a 230 Vac. Na figura 6.5 pode ser observado o esquema elétrico completo da placa de potência.

Tal como no desenho da placa de controlo foi necessário ter alguns cuidados no desenho da



placa de potência, uma vez que esta iria trabalhar a 230 V com correntes que podem atingir 10 A, optou-se por aumentar a distância entre as pistas do neutro e da fase, colocando-as em faces diferentes da placa. No desenho inicial, procurou-se que a posição dos componentes dedicados a cada saída ocupassem a menor área possível, tendo em conta que os conectores deviam estar perto dos bordos da placa. Os circuitos dedicados ao controlo da resistência de aquecimento e da bomba de circulação da água tiveram especial atenção, pois são as saídas que deverão aguentar correntes mais elevadas. Os pinos dos triac estão mais afastados de forma a usar pistas mais largas, a posição dos triac foi escolhida de forma a ficarem em linha para facilitar o uso de um dissipador de calor comum. Os restantes componentes, *shift-register*, LED e conectores foram colocados numa extremidade, de forma a facilitar a cablagem. Na figura 6.6 pode-se observar cada uma das faces da placa de potência.

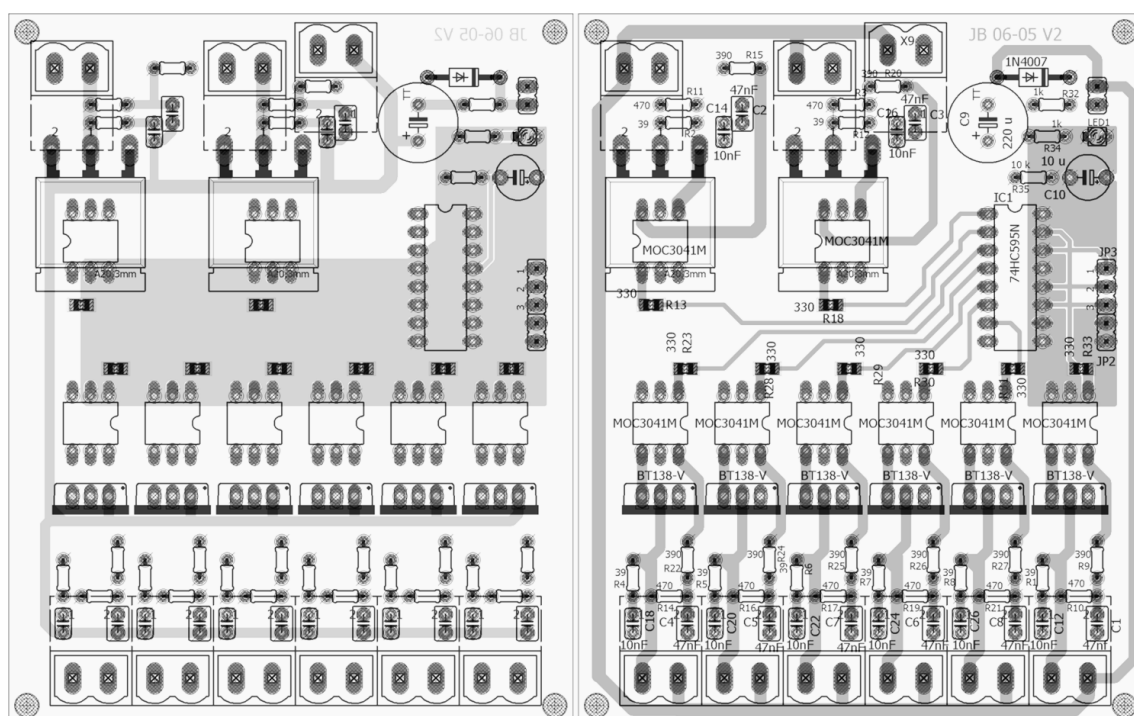


Figura 6.6: Placa de circuito impresso do módulo de potência, Bottom/Top

## 6.5 Alterações posteriores

Após o teste da placa na máquina deparou-se que existiam dois erros. O primeiro erro foi a inexistência do circuito de alimentação para o LED5 do painel de controlo. Como solução usou-se o circuito representado na figura 6.7, acionado diretamente pelo botão de corte geral da alimentação.

O segundo era que o Output Enable do shift-register (SF) estava conectado a GND, isto significa que o SF liberta instantaneamente os valores em memória para a saída, levantando assim um problema; quando o sistema inicia o SF contém lixo e uma vez que não

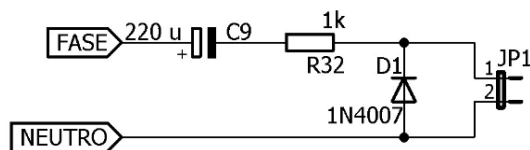


Figura 6.7: Circuito de alimentação do LED5

controlamos o Output Enable o valor que é tomado nas saídas do SF reflete-se na ativação dos componentes da máquina de lavar, este problema dura até que o valor das saídas esteja bem definido.

Este tempo dura desde que o sistema é alimentado até que a rotina *SPI Init()* é executada. É aqui que é limpo o lixo contido no SF sempre que este liga.

Uma possível solução para este problema era ligar uma saída do micro-controlador ao Output Enable do SF e impedir a propagação para as saídas do lixo contido no mesmo através de software. Porém para o protótipo optou-se por medir o tempo que demora até que leva as saídas estarem bem definidas e garantir por via do Hardware que o Output Enable do SF toma o valor lógico 1.

Após algumas medições calculou-se a média e com este valor assumiu-se que a constante de tempo duas vezes superior à medida. Com esta constante de tempo determinou-se o valor do condensador e da resistência a usar. Na figura 6.8 é possível observar a solução adotada para resolver este problema.

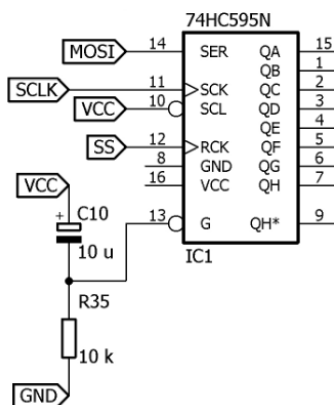


Figura 6.8: Circuito de correção do Output Enable

## Capítulo 7

# Interface com o utilizador

Neste capítulo é descrita a interface com o utilizador, quer a interface local através do painel de comando da máquina, quer a interface remota através de uma aplicação *Android*.

### Conteúdo

---

7.1	Interface local . . . . .	35
7.2	Aplicação para interface remota . . . . .	37
7.3	Comunicação microcontrolador - aplicação Android . . . . .	41
7.4	Ferramenta de desenvolvimento . . . . .	42
7.5	Desenvolvimento da aplicação . . . . .	43

---

### 7.1 Interface local

A figura 7.1 representa o novo painel da máquina.

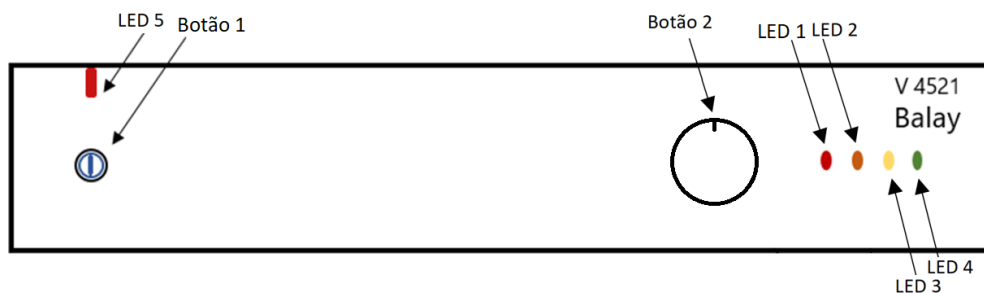


Figura 7.1: Painel de controlo da máquina

## Ligar a máquina

Certificar-se de que a máquina está ligada à rede elétrica e à rede de distribuição de água e que a porta da máquina está bem fechada. De seguida premir o botão 1 e verificar que luz azul do mesmo acende.

## Selecionar programa de lavagem

Quando a máquina é iniciada, entra no modo de escolha de programa de lavagem que, por omissão, é o programa P3: lavagem com carga completa. Para alterar o programa de lavagem basta girar o Botão 2. No painel são visíveis LEDs que se acendem e apagam conforme o programa selecionado:

- LED 1: Pré-lavagem
- LED 2: Lavagem intensiva com carga completa
- LED 3: Lavagem com carga completa
- LED 4: Lavagem com meia carga

Estes quatro programas correspondem aos programas implementados pelo controlador original. Uma vez selecionado o programa basta pressionar no Botão 2 para dar início ao programa.

Uma vez iniciado um programa, os LEDs mudam de significado, passando a indicar a percentagem de tempo que já decorreu no ciclo de lavagem.

- LED 1 – 0 a 25%
- LED 1 e LED 2 – 25 a 50%
- LED 1, LED 2 e LED 3 – 50 a 75%
- LED 1, LED 2, LED 3 e LED 4 – 75 a 100%

## Porta aberta

Caso a porta se abra durante o ciclo de lavagem o controlador da máquina para o ciclo de lavagem e memoriza o ponto da lavagem em que se encontra para que quando se voltar a fechar a porta se possa retomar o ciclo de lavagem no ponto em que foi interrompido. Como medida de segurança adicional, é cortada a alimentação de todos os componentes eletromecânicos da máquina. Esse corte é feito pelo próprio interruptor da porta.

## Terminar o ciclo de lavagem

O botão 1 interrompe a alimentação a todo o sistema.

## 7.2 Aplicação para interface remota

Foi decidido que a aplicação comunicaria com o micro-controlador da máquina através de *Bluetooth*, pelo facto de quase todos os *smartphones* de hoje em dia estarem equipados com *Bluetooth* e ser o método mais fácil de implementar do lado do controlador.

### Requisitos

Foram definidos como requisitos da App os seguintes:

- Aceder ao modo técnico
- Consultar o estado da máquina durante o ciclo de lavagem
- Escolher as características de lavagem
- Iniciar/Parar/retomar o ciclo de lavagem
- Permite conectar-se/desconectar-se da ligação com o micro-controlador
- Reiniciar o ciclo de lavagem

E como requisitos de comunicação do controlador:

- Enviar informação da máquina durante o ciclo de lavagem
- Receber comandos via Bluetooth

No desenvolvimento desta aplicação foi utilizada a ferramenta de desenvolvimento MIT App Inventor. Esta ferramenta gráfica facilita muito o desenvolvimento de Apps simples. Do lado do controlador foi utilizado o ambiente de trabalho Eclipse para a programação em C para a família AVR.

### Análise de requisitos

#### Actores

Como atores desta aplicação podem considerar-se dois: um utilizador normal e um técnico. Um utilizador normal tem acesso a algumas funcionalidades da aplicação podendo comandar a máquina (escolher as características de lavagem, consultar o estado da máquina; iniciar/parar/retomar o ciclo de lavagem). Um técnico tem acesso a todas as funcionalidades da aplicação.

#### Narrativas de utilização e casos de uso

Definiram-se as seguintes narrativas de utilização:

Tabela 7.1: Narrativas de utilização

Identificador	Descrição
US01	Como utilizador quero conectar/desconectar
US02	Como utilizador quero alterar as características de lavagem
US03	Como utilizador quero iniciar o ciclo de lavagem
US04	Como técnico aceder ao modo técnico
US05	Como utilizador quero parar/retomar o ciclo de lavagem
US05	Como utilizador quero saber o estado da máquina

Seguindo as narrativas de utilização identificadas para a aplicação, procurou-se identificar e caracterizar os requisitos dos utilizadores por forma a encontrar as soluções, tanto para os requisitos de informação como para os requisitos funcionais e não funcionais, e de comunicação com os utilizadores finais. Na tabela 7.2 estão descritos os casos de uso identificados.

Tabela 7.2: Casos de uso

Caso de uso	Prioridade	Descrição
UC01	Alta	Conectar/desconectar
UC02	Alta	Escolher programa de lavagem
UC02	alta	Escolher temperatura de lavagem
UC02	Alta	Escolher início deferido
UC03	Alta	Iniciar ciclo de lavagem
UC04	Baixa	Entrar no modo técnico
UC05	Alta	Parar ciclo de lavagem
UC05	Alta	Retomar ciclo de lavagem
UC06	Média	Aceder à interface Estado da máquina

## Interfaces

Na figura 7.2 e seguintes está descrita a visão geral de cada um dos tipos de atores bem como as diferentes interfaces.

**UI01: Menu** - Aqui o utilizador depara-se com o primeiro ecrã da aplicação onde pode escolher se deseja conectar-se ou desconectar-se da máquina, definir as características de lavagem, entrar no modo técnico, analisar o estado da máquina ou sair da App.

**UI02: Características do programa de lavagem** - Aqui o utilizador pode definir qual o programa de lavagem que pretende, a temperatura máxima de lavagem e o tempo (em horas) para iniciar o ciclo de lavagem.

**UI03: Estado da máquina** - Neste ecrã o utilizador pode obter informações acerca do estado em que a máquina se encontra durante o ciclo de lavagem, podendo também retomar ou reiniciar o ciclo de lavagem.



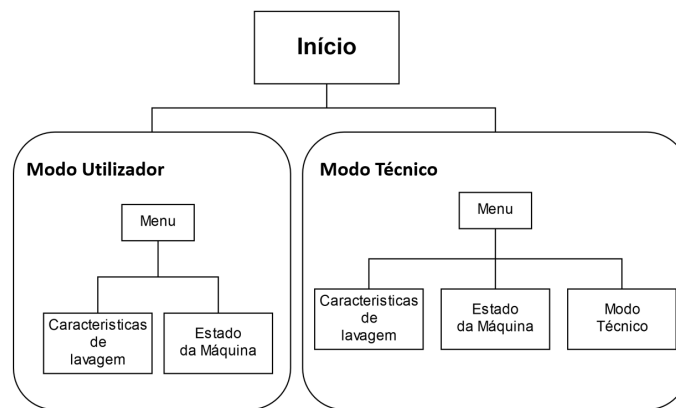


Figura 7.2: Visão geral

**UI04: Modo técnico** - Apenas o técnico, sabendo a palavra-passe, poderá aceder a este ecrã. Aqui pode ativar e desativar cada elemento da máquina, de modo a encontrar uma possível avaria. Não é possível ativar nenhum componente da máquina coma porta aberta. O técnico deverá ensaiar o fecho da porta na posição "fechada".

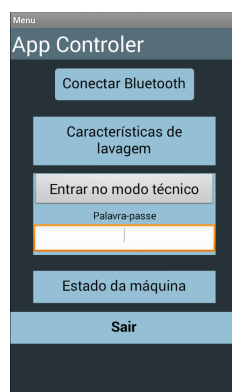


Figura 7.3: Menu principal

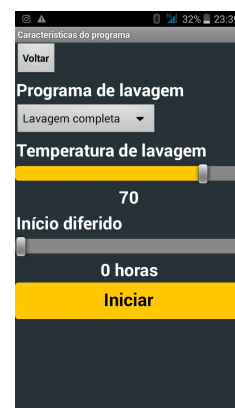


Figura 7.4: Características de programa

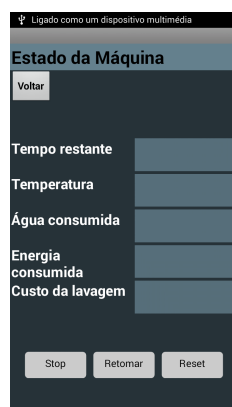


Figura 7.5: Estado da máquina

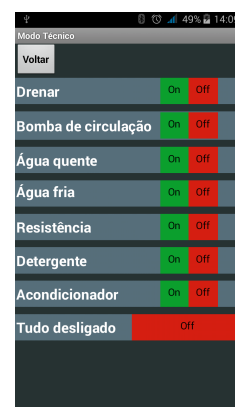


Figura 7.6: Modo técnico

## Modelo conceptual

Com a finalidade de perceber e modelar o problema a ser resolvido, foi construído um diagrama de classes *UML*, no qual foram representadas as entidades que permitiram atingir uma solução eficiente, e as respetivas associações entre elas. Para tentar criar o modelo da forma mais eficiente possível, procurou-se simplificar e decompor o problema em pequenas entidades.

O projeto utiliza uma aplicação que corre num smartphone com Bluetooth (cliente), e uma aplicação que corre no micro-controlador com o auxílio do módulo Bluetooth (servidor). A comunicação entre estas duas aplicações é feita através de USART, sendo as mensagens transmitidas segundo um protocolo exclusivo (Our Protocol).

No lado do cliente, a aplicação organiza-se em duas camadas:

- **Interface com utilizador:** Conjunto de classes responsável por interagir diretamente com o utilizador, recebendo e fornecendo informação;
- **Operação:** Conjunto de classes que permitem não só o tratamento da informação recebida e o seu envio para o utilizador, mas também a comunicação com o micro-controlador (servidor).

No lado do servidor existe apenas uma camada:

- **Operação:** que implementa a comunicação com o cliente.

O projecto segue portanto uma arquitetura cliente-servidor, encontrando-se a interface gráfica presente no lado do cliente onde se utilizou o MIT App Inventor 2 para criar a GUI (Grafical User Interface). Nesta parte do projeto existem diversas classes responsáveis por cada caso de uso. Existe uma classe responsável por cada janela com a qual o utilizador se depara no decorrer da utilização da aplicação e ainda uma classe *CommunicationsSlave* que é responsável pela parte de comunicações do cliente e também uma classe *BLSlave* (Business Logic Slave) que trata da comunicação vinda do controlador.

O programa está desenhado de modo a ter a seguinte hierarquia de funcionamento: o utilizador interage com a camada superior (GUI), que por sua vez interage com a camada *BLMaster*, que processa toda a informação; finalmente a classe *CommunicationsSlave* trata da comunicação através da USART. Desta forma garante-se uma boa escalabilidade, para quando for necessário adicionar alguma funcionalidade ou método.

O micro-controlador faz uso de um módulo Bluetooth para a comunicação com a aplicação *Android*, uma vez que esta será instalada em smartphones que já possuem esse tipo de conectividade.

Para que a informação sobre as características do ciclo de lavagem e sobre o estado da máquina seja trocada entre o microcontrolador e a aplicação *Android* foi definido o conjunto de mensagens descrito na secção seguinte.

### 7.3 Comunicação microcontrolador - aplicação Android

A comunicação entre o microcontrolador (servidor) e a aplicação *Android* (cliente) é feita via Bluetooth, recorrendo a um módulo compatível com a porta série do microcontrolador.

O microcontrolador foi programado de modo a que sofra um interrupção sempre que houver comunicação na porta série. Apenas a aplicação cliente é que toma a iniciativa de comunicar com o microcontrolador. As mensagens enviadas e recebidas estão codificadas de acordo com a tabela 7.3.

Tabela 7.3: Comandos enviados pelo cliente

Nome	Comando	Ação
Alloff	OF=0;	Desliga todos os componentes
CircBomb	CI=0;/CI=1;	Desliga/liga o motor de circulação da água
ColdWater	CW=0;/CW=1;	Fecha/abre a válvula de entrada de água fria
DrainBomb	DR=0;/DR=1;	Desliga ou liga a bomba de drenagem
Heater	HE=0;/HE=1;	Desliga/liga a resistência de aquecimento
HotWater	HW=0;/HW=1;	Fecha/abre a válvula de entrada de água quente
Info	G;	Pede informação sobre o estado da máquina
Settings	TTHHP;	Envia as características do ciclo de lavagem
Reset	RE;	Reinicia o ciclo de lavagem
Soap	SO=0;/SO=1;	Fecha ou abre a válvula do abrillhantar
Stop ou Start	ST = 0;/ ST = 1;	Para/retoma o ciclo de lavagem

No comando *Settings* é enviada uma string com todas as características do programa de lavagem. TT representa a temperatura, HH as horas de atraso no arranque e P o programa de lavagem escolhido (1, 2, 3 ou 4). A string é terminada por ";".

Da parte do controlador (servidor) é apenas necessário enviar informação sobre o estado da máquina: minutos para conclusão do programa de lavagem, flag de início de ciclo de lavagem, temperatura, água consumida, energia consumida. Assim como no comando *Settings* os dados são concatenados numa string, mas desta vez são separados por vírgulas. O formato é "F,MM,TT,CCC,EEE," em que F é a flag de início de ciclo de lavagem, MM representa os minutos que faltam até acabar o ciclo de lavagem, TT a temperatura, CCC a água consumida e EEE a energia consumida.

## 7.4 Ferramenta de desenvolvimento

A ferramenta utilizada disponibiliza dois painéis de desenvolvimento. No primeiro são criados os objetos disponíveis para a programação em blocos e é feito todo o design da aplicação. A figura 7.7 está evidenciado a azul um painel com a lista de objetos que podem ser utilizados na interface, organizadas por funções. No retângulo amarelo está a interface para onde são arrastados os objetos e posicionados onde se deseja. No retângulo verde estão os objetos utilizados na interface, neste é possível alterar o nome de cada objeto. No retângulo vermelho é possível alterar as propriedades de cada objeto.

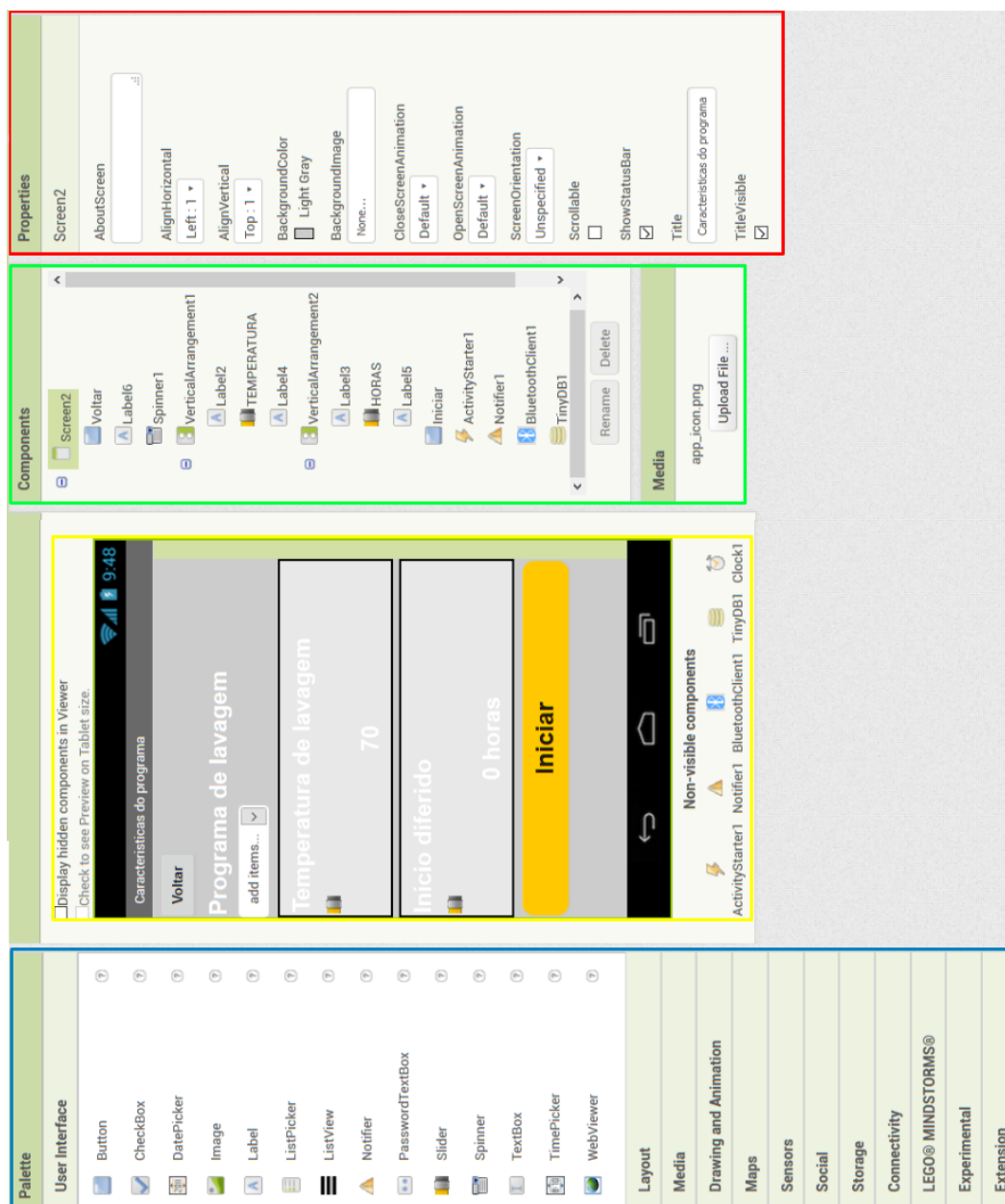


Figura 7.7: Desenho dos ecrãs

No segundo painel de desenvolvimento, pode ser desenvolvida a programação em blocos, aqui estão disponíveis os objetos criados no painel anterior e a partir desses é possível definir as ações que advêm do uso desses objetos.

A figura 7.8 mostra o segundo painel de desenvolvimento. No retângulo vermelho estão disponíveis os objetos e os selecionados no painel anterior. No retângulo verde estão disponíveis as possíveis ações dos objetos selecionados. No retângulo amarelo estão os blocos utilizados, onde é necessário interligá-los de forma lógica.

## 7.5 Desenvolvimento da aplicação

Inicialmente são inseridos os componentes na janela existente no lado esquerdo do primeiro painel, os mesmos estão organizados por grupos, (*User Interface, Layout, Sensors, Connectivity, Storage, etc*). De forma a ser possível ajustar verticalmente e horizontalmente os elementos é necessário inserir primeiro os objetos de *layout* e posteriormente são colocados os elementos chave da aplicação como os botões, caixas de texto, legendas, etc.

A interface inicial está representada na figura 7.3 e nesta pode-se observar que no design da primeira interface com o utilizador foram adicionados 3 botões ("Entrar no modo técnico", "Estado da máquina" e "Sair"), uma lista (Conectar) e 2 legendas (App máquina, Palavra-passe) e uma *PasswordTextBox*. Para além dos objetos visíveis foram adicionados outros não visíveis para o utilizador mas que são necessários para executar certas funções. Os objetos adicionados foram o *BluetoothClient*, que permite conectar e comunicar remotamente com outro dispositivo, uma *TinyDB* que é uma base de dados e por último foi adicionado um *Notifier* que permite enviar uma notificação ao utilizador da aplicação.

Depois de definidos os elementos do primeiro ecrã procedeu-se à programação em blocos. Quando a aplicação é iniciada o ecrã torna-se visível e de seguida é testado se o módulo Bluetooth do telemóvel está ativo através do objeto *BluetoothClient*, caso não esteja ativo com o objeto *Notifier* o utilizador é alertado do mesmo.

O objeto "Conectar" tem como função mostrar ao utilizador os dispositivos disponíveis e emparelhados com o *smartphone* com que poderá conectar-se. Depois o utilizador poderá escolher o dispositivo a conectar e de seguida é guardado endereço do mesmo na base de dados *TinyDB*.

Os botões "Características de lavagem" e "Estado da máquina" quando pressionados abrem uma nova interface, visíveis nas figuras 7.4 e 7.5 respetivamente e fecham a interface atual.

O botão "Entrar no modo técnico" verifica se a palavra-passe inserida na *PasswordTextBox* está correta, caso esteja fecha a presente interface e abre a interface da figura 7.6.

O botão "Sair" termina a aplicação.

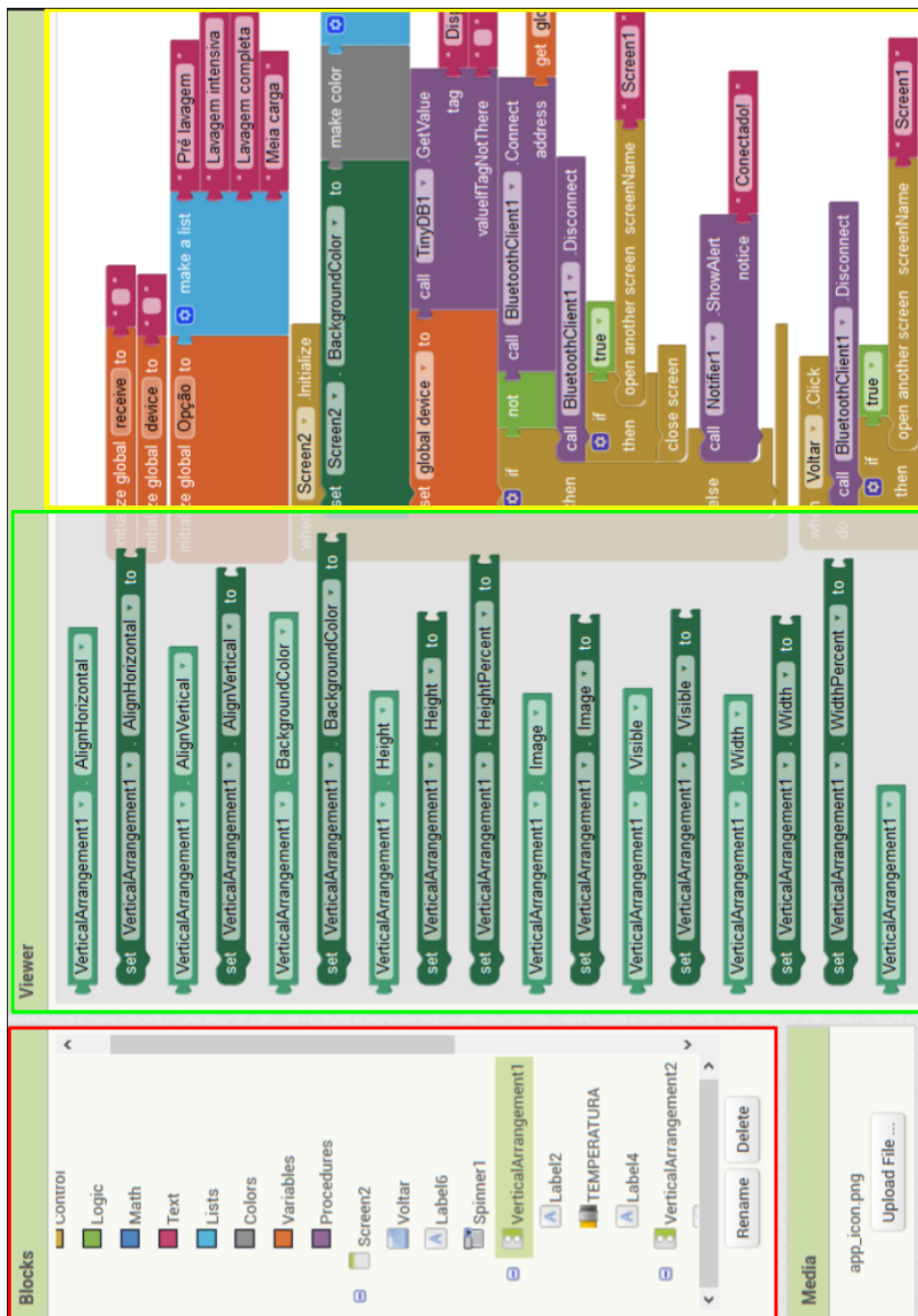


Figura 7.8: Painel da programação em blocos

Para a interface representada na figura 7.4 adicionaram-se os seguintes objetos, um *BluetoothClient*, a *TinyDB*, o *Notifier*, dois botões (Voltar e Iniciar), um *Spinner* este permite seleccionar uma das opções definidas, dois *Sliders* que permite escolher um valor de entre uma gama e cinco legendas.

Assim que a interface é iniciada é feita a ligação Bluetooth com o módulo do controlador, se esta ligação for mal sucedida a interface é encerrada e é aberta a anterior, caso seja bem

sucedida o utilizador recebe uma mensagem com o texto "Conectado!".

Como atuadores no painel de programação definiu-se o botão "Voltar" para abrir a interface anterior e encerrar a atual. O *Spinner* permite o utilizador escolher qual é o programa de lavagem desejado. O primeiro *Slider* tem como função permitir escolher qual a temperatura máxima a que se realizará o ciclo de lavagem, sendo possível escolher um valor entre 15 e 80 °C. A escolha supletiva por omissão é 70 °C. O segundo *Slider* permite escolher quantas horas mais tarde o ciclo de lavagem se vai iniciar. Sendo possível escolher valores entre 0 e 12 horas e tempo 0 como escolha supletiva por omissão. Para ambos os *Sliders* existe uma legenda que é atualizada para o valor do mesmo sempre que este é alterado.

Por último, o botão "Iniciar" quando pressionado testa se a conexão Bluetooth está ativada, caso esteja é enviado o comando "Settings" via Bluetooth para o controlador, depois de enviada é terminada a ligação remota e é fechada a presente interface e de seguida é feita abertura da interface inicial. A consulta do estado da máquina só é possível ser feita com o ciclo de lavagem a decorrer. Caso tente fazê-lo, a App vai automaticamente ser direcionada para o menu principal.

No desenvolvimento da interface "Modo técnico" representada na figura 7.6 foi necessário adicionar os seguintes objetos *BluetoothClient*, a *TinyDB*, o *Notifier*, um botão Voltar, dois botões para cada componente da máquina, um botão "Off" que desliga o componente em questão, outro botão "On" que liga, por último é adicionado outro botão, "Tudo desligado" que desliga todos os componentes de uma só vez.

Feito o design desta interface passa-se à programação por blocos. Quando a interface é iniciada começa-se por guardar o valor do endereço ao qual será feita a ligação remota que está na base de dados numa variável local, de seguida é feita a ligação e caso seja bem sucedida é feito o alerta ao utilizador que o dispositivo está conectado, caso contrário é dado o alerta oposto e de seguida é fechada esta interface e é aberta a interface inicial.

Uma vez que a conexão remota está garantida apenas é necessário programar os botões. Quando o botão "Voltar" é pressionado é enviada o comando "Alloff", termina a conexão Bluetooth, fecha a interface atual e abre a interface inicial.

O botão "Tudo desligado" envia o comando "Alloff". Os Botões restantes enviam os respetivos comandos.

A última interface pode ser visualizada na figura 7.5 e é constituída por oito objetos visíveis quatro botões e quatro legendas, os não visíveis são dois *Clock* com intervalo de 10 segundos e outro de 0.5 segundos, *BluetoothClient*, a *TinyDB* e o *Notifier*.

Quando esta interface é iniciada é feita a conexão Bluetooth e como acontece nas outras interfaces se for bem sucedida o utilizador recebe uma mensagem a alertar do sucesso da conexão, caso contrário recebe uma mensagem contrária e abre a interface inicial e fecha a atual.

O botão "Voltar" funciona como as interfaces anteriores, termina a conexão remota e abre a interface inicial. Os botões "Reset", "Retomar" e "Stop" enviam respetivamente o comando "Reset", "Start" e "Stop".

Os *Clock* são dois sinais de relógio de frequências diferentes usados para a sincronização da comunicação. O de frequência inferior, 0.1 Hz, envia o comando "Info", enquanto que o de frequência 2 Hz está constantemente a verificar se há bytes para receber, caso haja são guardados numa variável local. De seguida a string recebida é concatenada sempre que encontra o carácter ",", e os valores concatenados são guardados em variáveis locais. Quando se chegar ao final da string os dados disponibilizados nas legendas da interface são atualizados com novos valores.



## Capítulo 8

# Conclusão

### 8.1 Resultados e conclusões

Após os testes ao sistema desenvolvido, pode afirmar-se que foram cumpridos os objetivos propostos inicialmente.

O *hardware* do controlador da máquina foi desenvolvido com sucesso, quer enquanto protótipo laboratorial quer depois de passado ao estado de protótipo industrial em dois circuitos impressos separados. Foi também renovada e substancialmente simplificada toda a cablagem da máquina e adicionado um sensor de temperatura e um caudalímetro de modo a viabilizar algumas das novas funcionalidades tornando assim a máquina mais sofisticada.

O *software* desenvolvido para o controlador da máquina também verifica os objetivos propostos. Em modo de funcionamento normal, via comandos locais ou remotos, é possível:

- Replicar o ciclo de lavagem original
- Controlar a temperatura do ciclo de lavagem
- Medir o consumo de água e de energia elétrica
- Calcular o custo de lavagem
- Detetar anomalias no abastecimento de água e no estado da porta

Em modo de manutenção é possível controlar individualmente cada órgão mecânico da máquina, via comandos remotos.

A interface local com o utilizador foi modernizada e redesenhada. O utilizador pode agora obter mais informações do que a interface local original oferecia, nomeadamente o tempo restante para o fim do ciclo (LEDs 1 a 4 do painel) e informações relativas a eventuais anomalias (LED 5 do painel).

Foi desenvolvida uma aplicação Android para interface remota que permite substituir a operação local e ainda oferece funcionalidades adicionais como a escolha da temperatura,

de lavagem e o início diferido configurável bem como fornecer ao utilizador ainda mais informação sobre a lavagem, nomeadamente os consumos de água e de energia e o custo da lavagem. Essa mesma aplicação tem também um modo especial (modo de manutenção) em que é possível controlar individualmente cada órgão mecânico da máquina.

Pode portanto concluir-se que é viável o desenvolvimento de um controlador eletrónico para máquinas de lavar loiça de gama média e baixa, que as torne conectáveis, confirmando que o facto de se encontrarem essas características apenas em máquinas de gama alta se deve mais a razões de estratégia comercial do que a razões de ordem técnica.

A tese inicialmente apresentada de que esse desenvolvimento é possível recorrendo apenas a componentes eletrónicos facilmente acessíveis (COTS – Components off the Shelf) fica também aqui completamente verificada.

## 8.2 Trabalho futuro

Para além do que se conseguiu fica ainda muito que, com mais tempo, se poderia ter feito, pois houve um momento em que foi necessário tomar a decisão de parar. Listam-se de seguida diversas ideias que foram surgindo no decorrer do trabalho e que constituem sugestões para trabalho futuro.

Desde logo será de avaliar a utilização de tecnologia Wifi para a conexão remota. Desta forma a máquina poderá estar permanentemente conectada, sendo acessível de qualquer parte do mundo. Especial cuidado será necessário dedicar à questão da segurança.

Também poderão ser adicionados mais sensores, tornando a máquina mais versátil. Um sensor de turbidez ajudaria a controlar melhor o consumo de água, renovando-a apenas quando atingisse um determinado nível de sujidade. Um sensor de alerta de falta de sal no circuito de acondicionamento da água, e um de falta de liquido abrillantador seriam duas possíveis melhorias.

Para melhoria do software da máquina: implementar a possibilidade de parar e retomar o ciclo de lavagem através do painel local da máquina, tal como é possível na App; ampliar a deteção de anomalias de funcionamento tais como problemas na saída de água, no aquecimento e na circulação da água.

Por último, acrescentar uma entrada de água quente que, caso exista uma fonte de aquecimento de água mais eficiente do que a resistência interna da máquina, permitiria reduzir o custo de cada lavagem. Esta funcionalidade estava prevista desde início e é até muito simples de implementar mas, por motivos logísticos e operacionais relacionados com a rede de abastecimento de água quente do laboratório, não foi possível implementá-la.

# Referências

- [1] Deco proteste. Máquinas de lavar loiça.  
<https://www.deco.proteste.pt/eletrodomesticos/maquinas-lavar-loica>, 2017.  
Consultado em 02/03/2017.
- [2] Deco proteste. Telemóveis.  
<https://www.deco.proteste.pt/tecnologia/telemoveis>, 2017. Consultado em 04/03/2017.
- [3] Deco proteste. Consumo de água na lavagem com máquina vs manual.  
<https://www.deco.proteste.pt/eletrodomesticos/maquinas-lavar-loica/dicas/maquina-de-lavar-loica-poupe-da-compra-ao-fim-de-vida>, 2017.  
Consultado em 28/04/2017.
- [4] Homestuffworks. How dishwashers work.  
<http://home.howstuffworks.com/dishwasher.html>, 2017. Consultado em 20/04/2017.
- [5] ExplainThatStuff. How do dishwashers work.  
<http://www.explainthatstuff.com/dishwashers.html>, 2017. Consultado em 20/04/2017.
- [6] BOSCH. Dishwasher.  
<http://skewred.com/42-bosch-exxcel-dishwasher-parts-diagram/>, 2017.  
Consultado em 26/04/2017.
- [7] Austin Hughes. *Electric Motors and Drives*. Elsevier Ltd, 2006.
- [8] Wikipédia. Ion-exchange resin.  
[https://en.wikipedia.org/wiki/Ion-exchange\\_resin](https://en.wikipedia.org/wiki/Ion-exchange_resin), 2017. Consultado em 26/04/2017.
- [9] UK whitegoods. How Water Softening Systems Work.  
<http://www.ukwhitegoods.co.uk/help/fix-it-yourself/dishwasher-self-help/2711-water-softening-systems>, 2017. Consultado em 26/04/2017.



## Anexo A

# Código do Datalogger

```
1 //Configuracao da porta serie usada para comunicar com o computador
2 #define UBBR_VAL 207
3 void init_usart(void) {
4     UBRROH = (uint8_t)(UBBR_VAL>>8);
5     UBRROL = (uint8_t) UBBR_VAL;
6     UCSROA = (1<<U2X0);          // Double speed
7     UCSROC = (3<<UCSZ00)         // 8 data bits
8         | (0<<UPM00)             // no parity
9         | (0<<USBS0);           // 1 stop bit
10    UCSROB = (1<<TXEN0);          // Activate Tx only
11 }

1 //Definiu-se um periodo de 1 segundo para a aquisicao e transmissao
  de dados. Configuracao do timer 1 usada e a seguinte.
2 #define T1COUNT 65536-15625
3 void T1_init(void){
4     TCCR1B = 0;                  //para o TC1
5     TCCR1A = 0;                  //normal mode
6     TCNT1 = T1COUNT;           //valor inicial do contador
7     TIMSK1 = 0x01;              //enable overflow interrupt
8     TCCR1B = 0x05;              //iniciar o TC1 com 1
    prescaler de 64 => T = 64/16MHz = 4us
9 }

1 ISR(TIMER1_OVF_vect){
2     TCNT1 = T1COUNT;           //repoe a contagem
3     pino1 = PINB;
4     pino2 = PIND;
5     temperatura = adc_read(3)/9.3; //leitura da temperatura
6     contador = impulsos;         //consumo de agua do momento
7     print = 1;                   //flag par impressao
8 }
```

```

1 //Configuracao da interrupcao externa por forma a contar os
  impulsos gerados pelo caudalimetro
2 void init_int0(void) {
3     DDRD = DDRD & ~(1<<INT0);           // INTO as input
4     PORTD = PORTD | (1<<INT0);          // Activate internal
  pull-up resistor
5     EICRA = 0b00000010;                 // INTO active on
  falling edge
6     EIMSK = (1<<INT0);                 // INTO enabled
7     sei();                             // Global INT enable
8 }
9

1 //Dividindo o numero de impulsos por 394 impulso temos os litros de
  agua consumidos
2 ISR(INT0_vect) {
3     impulsos ++;
4 }

1 //Configuracao do pino analogico para leitura da temperatura.
2     ADMUX = (3<<REFS0)|(3<<MUX0);       // Vref=1.1V;
  Channel 3
3     DIDR0 = (1<<3);                     // Deactivate
  digital IO at Ch 3
4     ADCSRA = (7<<ADPS0)|(1<<ADEN);      // Enable ADC with
  128 prescaler

1 //Na rotina seguinte da-se a conversao do valor lido e e retornado
  o valor da temperatura.
2 uint16_t adc_read(uint8_t ch){
3     ADMUX = (ADMUX & 0xF0) | (ch & 0x0F); // Specify channel
4     ADCSRA |= (1<<ADSC);                 // Start conversion
5     while (ADCSRA & (1<<ADSC));
6     return ADC;
7
8 //Por fim para sabermos a temperatura a que correta basta dividir
  esse valor por 9.3.
9 }

1 // Uma vez que os dados terao de ser enviados pela porta serie
  criou-se as seguintes rotinas. A primeira faz a impressao de
  apenas um caracter e a segunda faz impressao de um numero ate
  65536.
2
3 void myputchar(uint8_t c){
4     while((UCSROA & (1<<UDRE0)) == 0);

```

```

5   UDR0 = c;
6   }
7   void myprintdeci(uint16_t x){
8       if (x == 0) myputchar('0');
9       else {
10          uint16_t y = 0, z = x;
11          uint8_t n = 0;
12          while (z != 0){
13              y = y*10 + z%10;
14              z = z/10;
15              n++;
16          }
17          while (n != 0){
18              myputchar(y%10 + '0');
19              y = y/10;
20              n--;
21          }
22      }
23  }

1  // Nesta rotina sao definidos os pinos que serao usados como
   entradas ou saidas.
2  void init_io(void)
3      DDRB = 0b00000000;           // PB as input
4      PORTB = 0b11111111;          // Enable all pull-ups
5      DDRD = DDRD & 0b11001111;    // PD5 and PD6 as inputs
6      PORTD = 0b00110000;          // Enable pull-ups
7      DDRC = DDRC & 0b11111100;    // PC0 e PC1 as inputs
8      PORTC = 0b00001000;          // Enable pull-up
9  }

1  // No programa principal chamam-se as rotinas inicializacao dos
   pinos de entrada e saida, de configuracao do pino analogico, da
   porta serie, do timer1 e da interrupcao externa. Dentro do ciclo
   infinito faz-se o envio dos dados via porta serie sempre que a
   flag de impressao e ativada. Os dados sao enviados para o
   computador pela seguinte ordem: acondicionamento, abrlhantador,
   valvula de entrada de agua, resistencia, bomba de circulacao,
   consumo de agua, bomba de drenagem, temperatura.
2
3  int main (void){
4      init_io();
5      init_ADC();
6      init_usart();
7      T1_init();
8      init_int0();

```

```
9
10  while(1) {
11      if(print) {
12          uint8_t i;
13          for(i = 32; i > 1 ; i = i>>1){
14              if(pino1 & i) myputchar('1');
15              else myputchar('0');
16              myputchar(',');
17          }
18          myprintdeci(contador);
19          myputchar(',');
20
21          if(pino1 & 0x01) myputchar('1'); else myputchar('0');
22          myputchar(',');
23
24          if(pino2 & 0x20) myputchar('1'); else myputchar('0');
25          myputchar(',');
26
27          myprintdeci(temperatura);
28          myputchar('\n');
29          print = 0;
30      }
31  }
32 }
```



## Anexo B

# Código do Controlador

```
1 // rotina de leitura da temperatura atraves do pino analogico PC0
2 #define ADC_PIN 0
3 uint16_t adc_read(void){
4     ADCSRA |= _BV(ADEN);
5     ADMUX &= 0xf0; // Specify channel
6     ADCSRA |= _BV(ADSC); // Start conversion
7     while ( (ADCSRA & _BV(ADSC)) );
8     return ADC;
9 }
10
11 // rotinha de inicializacao de todos os pinos digitais do
12 // controlador e das duas interrupcoes externas dos pinos PC0 e PD1
13 void init_io(void){
14     DDRB |= ~(1<<PB4); // PB4 as inputs
15     PORTD |= (1<<PB4); // with pull-ups
16
17     //Port D
18     DDRD &= ~(7<<2); // PD2,PD3,PD4 as
19     // inputs
20     PORTD |= (7<<2); // with pull-ups
21
22     //Port C
23     DDRC |= (1<<PC2); //PC2 as outputs
24     DDRC |= ~(1<<PC1) | ~(1<<PC3); //PC1 as inputs
25     PORTC |= (1<<PC1)|(1<<PC3); //with pull-ups
26
27     EICRA = EICRA | (2<<2) // INT1 mode 2 (
28     // falling edge)
29     | (1<<0); // INT0 mode 1 (both edges)
30     // IMPORTANT!
```

```

28  EIMSK = EIMSK | (1<<INT0)           // enable INT0
29      | (1<<INT1);                   // enable INT1
30  }
31
32  // rotina de configuracao da interrupcao externa PCINT9 para contar
    os impulsos gerados pelo caudalimetro
33  void init_PCINT9(void){
34      PCICR  |= PCIE1;
35      PCMSK1 |= PCINT9;
36      PCICR  |= PCIE1;
37      sei();
38  }
39  // rotina de configuracao da interrupcao interna, esta configurada
    para ser chamada de 1 em 1 segundo
40  #define T1COUNT 65536-15625
41  void T1_init(void){
42      TCCR1B = 0;                      //para o TC1
43      TCCR1A = 0;                      //normal mode
44      TCNT1  = T1COUNT;              //valor inicial do
    contador
45      TIMSK1 = 0x01;                  //enable overflow
    interrupt
46      TCCR1B = 0x05;                  //iniciar o TC1 com 1
    prescaler de 1024 => T = 1024/16MHz = 1/15625
47  }

1  // rotina de configuracao da comunicacao SPI que ira comunicar com
    o shift-register do modulo de potencia
2  void SPI_Init(void){
3      /* Set SS, MOSI and SCK output */
4      DDRB |= (1<<PB2)|(1<<PB3)|(1<<PB5);
5      //DDRB = ~(1<<PB4);
6      /* Enable SPI, Master, set clock rate fck/16 |(1<<DORD)*/
7      SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
8
9      cData = 0x00000000;
10     SPI_Transmit(cData);
11 }
12
13 // rotina de transmissao de dados
14 void SPI_Transmit(char cData){
15     // Inicio da transmissao
16     PORTB = PORTB & ~(1<<PB2);
17     SPDR = cData;
18
19     // Espera pela finalizacao da transmissao */

```

```

20  while(!(SPSR & (1<<SPIF)));
21  PORTB = PORTB | (1<<PB2);
22  }

1  // rotina de configuracao da porta serie, atraves desta porta e
   feita comunicacao com o modulo Bluetooth que comunica com a App
   Android
2
3  #define UBBR_VAL 207
4  void init_usart(void) {
5      UBRROH = (uint8_t)(UBBR_VAL>>8);
6      UBRROL = (uint8_t) UBBR_VAL;
7      UCSROA = (1<<U2X0);           // dobro da velocidade
8      UCSROC = (3<<UCSZ00)           // 8 data bits
9              | (0<<UPM00)           // sem bits de paridade
10             | (0<<USBS0);           // 1 stop bit
11      UCSROB = (1<<RXEN0)|(1<<TXEN0); // Ativar o Rx e o Tx
12  }
13
14 // rotina que envia um caracter pela porta serie
15 void myputchar(uint8_t c){
16     while((UCSROA & (1<<UDRE0)) == 0);
17     UDR0 = c;
18 }
19
20 // rotina que envia um numero com dimensao de 0 a 65536 pela porta
   serie
21 void myprintdeci(uint16_t x){
22     if (x == 0) myputchar('0');
23     else {
24         uint16_t y = 0, z = x;
25         uint8_t n = 0;
26         while (z != 0){
27             y = y*10 + z%10;
28             z = z/10;
29             n++;
30         }
31         while (n != 0){
32             myputchar(y%10 + '0');
33             y = y/10;
34             n--;
35         }
36     }
37 }
38
39 // rotina que envia uma string pela porta serie

```

```
40 void myprint(char *p) {
41     char c;
42     while ((c = *p++) != 0)
43         myputchar(c);
44 }

1 // rotina de interrupcao ativada pelo pino INT0, a sua ativacao
   esta ligada ao codificador incremental (rotary encoder) do
   painel da maquina e sempre que o utilizador altera a posicao do
   codificar a interrupcao e ativada
2 ISR(INT0_vect){
3     ppa = PIND & 0b00010100;    //present state of the encoder signals
4     if ((ppo==0b00000000)&&(ppa==0b00010100)) value++;
5     else if ((ppo==0b00000100)&&(ppa==0b00010000)) value--;
6     ppo=ppa;    //update old state of the encoder
7     value = value % 16;    // 0 < prog < 15
8 }
9
10 // rotina de interrupcao ativada pelo pino INT1, a ativacao esta
   ligada ao botao de pressao do painel da maquina. Quando ativada
   pela primeira vez da inicio ao ciclo de lavagem ao
11 ISR(INT1_vect){
12     if(Flag_botao){
13         switch (programa){
14             case 0:
15                 T_total = 460;
16                 break;
17             case 1:
18                 T_total = 5472;
19                 break;
20             case 2:
21                 T_total = 5077;
22                 break;
23             case 3:
24                 T_total = 4682;
25                 break;
26             default:
27                 T_total = 5077;
28                 break;
29         }
30         Flag_start = 1;
31         Flag_botao = 0;
32     }
33 }
34
```

```
35 // rotina de interrupcao ativada pelo pino PCINT9, e a rotina que
    conta quantos impulsos o caudalimetro gerou
36 ISR(PCINT9_vect) {
37     impulsos++;
38 }
39 // rotina de interrupcao interna, aqui faz-se a verificacao e
    controlado o ciclo de lavagem
40 ISR(TIMER1_OVF_vect){
41     TCNT1 = 65536-15625;    // repoe a contagem
42
43
44     if(!Flag_standby){      // esta em espera que o
        inicio diferido acabe
45         if(Inicio_diferida <= 1){    // da inicio ao ciclo de
            lavagem
46             tempo = 0;
47             counter = 0;
48             Flag_start = 1;
49             Flag_standby = 1;
50             return;
51         }
52         Inicio_diferida--;
53         return;
54     }
55     if(Flag_start){        // esta dentro do ciclo de
        lavagem
56         tempo++;
57         counter++;
58         debug();           // verifica se ha abastecimento de agua
59         check_door();      // verifica se a porta esta fechada
60         Check_temperatura(); // verifica se a temperatura nao
            ultrapassou a desejada
61         return;
62     }
63 }
64
65 // rotina de interrupcao da porta serie, sempre que ha comunicacao
    do exterior para o microcontrolador esta rotina e ativada. Faz-
    se uma pre filtragem dos comandos.
66 ISR(USART_RX_vect){
67     volatile unsigned char  RecByte;
68     static unsigned int     uc_process = 0;    // flag de comando a
        ser processado
69     static unsigned char    uc_buffer[16];
70     static unsigned int     uc_n = 0;        // indice do buffer
```

```

71  static unsigned int    uc_parameter = 0;    // parametro do comando
      (1 ativar 0 desligar)
72  char                  uc_comand[2];          // comando
73
74  RecByte = UDR0;
75  uc_buffer[uc_n++] = RecByte;
76
77  if (RecByte == ';'') {                      // sinaliza o fim do comando
78      uc_process = 1;
79  }
80
81  if(uc_process){
82      if(uc_buffer[0] == 'G'){
83          dump();
84      }else if('I' == uc_buffer[0]){
85          set_carac(uc_buffer);                // define as
      características do programa de lavagem
86      }else{
87          setComand(uc_buffer, uc_comand);      // define o
      comando recebido
88          uc_parameter = setparameter(uc_buffer); // define o
      parametro do comando
89          update_state(uc_parameter, uc_comand); // aplica o
      comando
90          SPI_Transmit(cData);                  // envia o
      comando via SPI
91          uc_parameter = 0;                      // reinicia o
      parametro
92      }
93      uc_process = 0;                          // reinicia a
      flag
94      uc_n = 0;
95      for (int i=0; i<16; i++)                  // limpa o
      buffer
96          uc_buffer[i] = 0;
97  }
98  return;
99 }

1 void set_carac(unsigned char uc_buffer[]){
2     int i = 1;
3     while(uc_buffer[i] != ';'') {              // conta o numero de
      caracteres do comando
4         i++;
5     }
6

```

```
7   Temperatura = (uc_buffer[1]-48)*10 + uc_buffer[2]-48;
      // define a temperatura de lavagem
8
9   if(6 == i){
10      Inicio_diferida = ((uc_buffer[3]-48)*10 + uc_buffer[4]-48)
      *3600;          // define o inicio diferido
11      programa = uc_buffer[5]-49;
      // define o ciclo de lavagem
12      i = 1;
13      Flag_standby = 0;
      // permite iniciar a lavagem
14      return;
15   }else if(5 == i) {
16      Inicio_diferida = (uc_buffer[3]-48)*3600;
17      programa = uc_buffer[4]-49;
18      switch (programa){          // define a duracao do ciclo de
      lavagem
19          case 0:
20              T_total = 460;
21              break;
22          case 1:
23              T_total = 5472;
24              break;
25          case 2:
26              T_total = 5077;
27              break;
28          case 3:
29              T_total = 4682;
30              break;
31          default:
32              T_total = 5077;
33              break;
34      }
35      i = 1;
36      Flag_standby = 0;
37      return;
38   }
39 }
40 void setComand(unsigned char uc_buffer[], char *uc_comand){    //
      define o comando recebido
41   if(uc_buffer[0] == 'o' || uc_buffer[0] == 'O'){
42       uc_comand[0] = 'O';
43       if(uc_buffer[1] == 'f' || uc_buffer[1] == 'F'){
44           uc_comand[1] = 'F';
45           return;
```

```
46     }
47 }
48 if(uc_buffer[0] == 'd' || uc_buffer[0] == 'D'){
49     uc_comand[0] = 'D';
50     if(uc_buffer[1] == 'r' || uc_buffer[1] == 'R'){
51         uc_comand[1] = 'R';
52         return;
53     }else if(uc_buffer[1] == 'u' || uc_buffer[1] == 'U'){
54         uc_comand[1] = 'U';
55         return;
56     }else if(uc_buffer[1] == 'e' || uc_buffer[1] == 'E'){
57         uc_comand[1] = 'E';
58         return;
59     }else{
60         return;
61     }
62
63
64 }else if(uc_buffer[0] == 'c' || uc_buffer[0] == 'C'){
65     uc_comand[0] = 'C';
66     if(uc_buffer[1] == 'I' || uc_buffer[1] == 'i'){
67         uc_comand[1] = 'I';
68         return;
69     }else if(uc_buffer[1] == 'w' || uc_buffer[1] == 'W'){
70         uc_comand[1] = 'W';
71         return;
72     }else{
73         return;
74     }
75
76 }else if(uc_buffer[0] == 's' || uc_buffer[0] == 'S'){
77     uc_comand[0] = 'S';
78     if(uc_buffer[1] == 'o' || uc_buffer[1] == 'O'){
79         uc_comand[1] = 'O';
80         return;
81     }else if(uc_buffer[1] == 't' || uc_buffer[1] == 'T'){
82         uc_comand[1] = 'T';
83         return;
84     }else{
85         return;
86     }
87
88 }else if(uc_buffer[0] == 'p' || uc_buffer[0] == 'P'){
89     uc_comand[0] = 'P';
90     return;
```



```
91 }else if(uc_buffer[0] == 'h' || uc_buffer[0] == 'H'){
92     uc_comand[0] = 'H';
93     if(uc_buffer[1] == 'E' || uc_buffer[1] == 'e'){
94         uc_comand[1] = 'E';
95         return;
96     }else if(uc_buffer[1] == 'w' || uc_buffer[1] == 'W'){
97         uc_comand[1] = 'W';
98         return;
99     }
100     return;
101 }else if(uc_buffer[0] == 'r' || uc_buffer[0] == 'R'){
102     uc_comand[0] = 'R';
103     return;
104 }else if(uc_buffer[0] == 't' || uc_buffer[0] == 'T'){
105     uc_comand[0] = 'T';
106     return;
107 }else if(uc_buffer[0] == 'A' || uc_buffer[0] == 'a'){
108     uc_comand[0] = 'A';
109     return;
110 }
111 return;
112 }
113
114 int setparameter(unsigned char uc_buffer[]){           // define o
115     parametro recebido
116     int i = 2;
117     int uc_parameter = 0;
118     while(uc_buffer[i] != ';'){
119         myputchar(uc_buffer[i]);
120         if(48 <= uc_buffer[i] && uc_buffer[i] <= 57 && uc_buffer[i] !=
121             '='){
122             uc_parameter = uc_buffer[i]-48;
123             return uc_parameter;
124         }else
125             i++;
126     }
127     return uc_parameter;
128 }
129 void update_state(unsigned int uc_parameter, char uc_comand[]){
130     // aplica o parametro
131     myprintdeci(uc_parameter);
132     volatile char cData_old = cData;
133     uint8_t temp = 0;
134     temp = adc_read();
135 }
```

```
133  if(uc_parameter <= 9 && uc_parameter >= 1){
134      if(uc_comand[0] == 'D' && uc_comand[1] == 'R'){
135          cData = cData | 0b01000000;
136          uc_parameter = 0;
137          return;
138      }else if(uc_comand[0] == 'C' && uc_comand[1] == 'W'){
139          cData = cData | 0b10000000;
140          uc_parameter = 0;
141          return;
142      }else if(uc_comand[0] == 'C' && uc_comand[1] == 'I'){
143          cData = cData | 0b00000100;
144          uc_parameter = 0;
145          return;
146      }else if(uc_comand[0] == 'S' && uc_comand[1] == 'O'){
147          cData = cData | 0b00010000;
148          uc_parameter = 0;
149          return;
150      }else if(uc_comand[0] == 'H' && uc_comand[1] == 'W'){
151          cData = cData | 0b00001000;
152          uc_parameter = 0;
153          return;
154      }else if(uc_comand[0] == 'H' && uc_comand[1] == 'E'){
155          cData = cData | 0b00000010;
156          uc_parameter = 0;
157          return;
158      }else if(uc_comand[0] == 'T'){
159          Temperatura = uc_parameter;
160          uc_parameter = 0;
161          return;
162      }else if(uc_comand[0] == 'P'){
163          programa = uc_parameter;
164          uc_parameter = 0;
165          return;
166      }else if(uc_comand[0] == 'A'){
167          cData = cData | 0b00100000;
168          uc_parameter = 0;
169          return;
170      }else if(uc_comand[0] == 'S' && uc_comand[1] == 'T'){
171          Flag_start = 1;
172          cData_old = cData;
173          cData = 0x00000000;
174          SPI_Transmit(cData);
175          uc_parameter = 0;
176          return;
177      }
```

```

178
179 }else if(uc_parameter == 0){
180     if(uc_comand[0] == 'D' && uc_comand[1] == 'R'){
181         cData = cData & 0b10111111;
182         return;
183     }else if(uc_comand[0] == 'C' && uc_comand[1] == 'W'){
184         cData = cData & 0b01111111;
185         return;
186     }else if(uc_comand[0] == 'C' && uc_comand[1] == 'I'){
187         cData = cData & 0b11111011;
188         return;
189     }else if(uc_comand[0] == 'S' && uc_comand[1] == 'O'){
190         cData = cData & 0b11101111;
191         return;
192     }else if(uc_comand[0] == 'H' && uc_comand[1] == 'W'){
193         cData = cData & 0b11110111;
194         return;
195     }else if(uc_comand[0] == 'H' && uc_comand[1] == 'E'){
196         cData = cData & 0b11111101;
197         return;
198     }else if(uc_comand[0] == 'P'){
199         programa = uc_parameter;
200         return;
201     }else if(uc_comand[0] == 'A'){
202         cData = cData & 0b11011111;
203         return;
204     }else if(uc_comand[0] == 'S' && uc_comand[1] == 'T'){
205         if(Flag_start == 1){
206             SPI_Transmit(cData_old);
207             Flag_start = 0;
208         }
209         return;
210     }else if(uc_comand[0] == 'O' && uc_comand[1] == 'F'){
211         cData = 0b00000000;
212         return;
213     }
214 }
215 if(uc_comand[0] == 'R'){
216     Flag_start = 1;
217     state = 0;
218     return;
219 }
220 return;
221 }

```

```

1 // define qual e o programa escolhido pelo

```

```

2 void check_programa(void){                                     utilizador, neste caso
    definido pelo painel local
3   if (value != old_value) {                                     // atualiza os LEDs do
    painel local, informando o utilizador de qual e o programa
    selecionado
4     if(0 <= value && value < 4){
5       programa = 0;
6       PORTD = PORTD | 0b01000000;    //ligar PD6
7       PORTB = PORTB & 0b111100;    //Desligar os outros
8       PORTD = PORTD & 0b01111111;    //Desligar os outros
9     }else if(4 <= value && value < 8){
10      programa = 1;
11      PORTD = PORTD | 0b10000000;    //ligar PD7
12      PORTB = PORTB & 0b111100;    //Desligar os outros
13      PORTD = PORTD & 0b10111111;    //Desligar os outros
14    }else if(8 <= value && value < 12){
15      programa = 2;
16      PORTB = PORTB | 0b0000001;    //ligar PB0
17      PORTB = PORTB & 0b111101;    //Desligar os outros
18      PORTD = PORTD & 0b00111111;    //Desligar os outros
19    }else if(12 <= value && value < 16){
20      programa = 3;
21      PORTB = PORTB | 0b0000010;    //ligar PB1
22      PORTB = PORTB & 0b111110;    //Desligar os outros
23      PORTD = PORTD & 0b00111111;    //Desligar os outros
24    }
25    old_value = value;
26  }
27 }
28
29 // atualiza o painel local da maquina durante o ciclo de lavagem.
    Da a informacao ao utilizador de quanto tempo falta para
    terminar o ciclo de lavagem
30 void painel_local(void){
31   int percentagem = 0;
32   percentagem = (tempo / T_total)*100;
33   if(0 <= percentagem && percentagem < 25){
34     PORTD = PORTD | 0b01000000;    //ligar PD6
35     return;
36   }else if(25 <= percentagem && percentagem < 50){
37     PORTD = PORTD | 0b10000000;    //ligar PD7
38     return;
39   }else if(50 <= percentagem && percentagem < 75){
40     PORTB = PORTB | 0b0000001;    //ligar PB0
41     return;

```

```
42 }else if(75 <= percentagem && percentagem <= 100){
43     PORTB = PORTB | 0b000010;    //ligar PB1
44     return;
45 }
46 return;
47 }
48 }

1 // rotina de verificacao do estado da porta, caso esta esteja
  fechada esta para o ciclo de lavagem, guardando o estado em que
  estava para se retome o estado em que estava quando se fechar a
  porta
2 void check_door(void){
3     volatile static unsigned int Flag_porta = 0;
4     volatile char cData_old;
5
6     if(!Flag_porta && (PIND & 32)){           //porta aberta
7         Flag_start = 0;                       // para o ciclo de lavagem
8         cData_old = cData;                     //guarda o estado atual
9         cData = 0x00000000;
10        Flag_porta = 1;                       // porta aberta
11        return;
12    }else if(!(PIND & 32) && Flag_porta){ //porta fechada
13        Flag_porta = 0;
14        cData = cData_old;
15        SPI_Transmit(cData);                   // repoe o estado
16        Flag_start = 1;                       // continua o ciclo
17        return;
18    }
19    return;
20 }

1 // rotina que permite que a resistencia seja ativada.
2 int Enable_heater(void){
3     if(state == 17 || state == 18 || state == 19 || state == 22 ||
4        state == 23){    //estados em que se liga a resistencia
5         return 1;
6     }else
7         return 0;
8 // rotina em que e verificada a temperatura da maquina. Aqui e lido
  o valor da temperatura, caso este iguale ou ultrapasse o valor
  desejado e desligada a resistencia de aquecimento. Quando a
  temperatura baixar 2 graus da temperatura desejada esta ativa a
  resistencia
8 void Check_temperatura(void){
9     volatile unsigned int    Flag_temp;
```

```

10  uint8_t temp = 0;
11
12  temp = adc_read();
13  Flag_temp = Enable_heater();
14
15  if(Flag_temp){
16      if(temp/10 >= Temperatura){
17          cData = cData & 0b11110111;
18          SPI_Transmit(cData);
19          return;
20      }else if((temp/10) <= (Temperatura-2)){
21          cData |= 0b00001000;
22          SPI_Transmit(cData);
23          return;
24      }
25  }
26  return;
27 }
28 }

1 // rotina que deteta problemas no abastecimento de agua. Caso haja
   e terminado o ciclo de lavagem e informa o utilizador atraves da
   ativacao do LED5 do painel local
2 #define tempo_enchimento 60 // tempo maximo de enchimento de 1
   deposito
3 void debug (void){
4     if(((state == 2) | (state == 4) | (state == 8) | (state == 10) |
        (state == 31) | (state == 33)) && counter > tempo_enchimento){
5         state = 27;           // vai o penultimo estado (drena a agua
        existente)
6         PORTC = PORTC | 0b00000010; // liga o LED5 - problema de
        abastecimento de agua
7     }
8     return;
9 }
10 // rotina que repoe as condicoes iniciais e executa sempre que o
    ciclo de lavagem e terminado
11 void set_initial_conditions(void){
12     state = 0;
13     counter = 0;
14     programa = 3;
15     Temperatura = 70;
16     tempo = 0;
17     Flag_start = 0;
18     Flag_standby = 1;
19     Flag_botao = 1;

```

```
20   notifier();
21 }

1  // rotina de transmissao da informacao sobre o estado da maquina,
   envia em primeiro lugar a Flag_start, depois os minutos
   restantes para terminar o ciclo de lavagem, depois a temperatura
   atual da maquina, depois o consumo de agua e por fim o custo
   monetario da lavagem. Esta informacao vai separada por virgulas
   para facilitar a concatenacao da informacao na App
2  void dump(void){
3   uint8_t temp = 0;
4   temp = adc_read();
5
6   unsigned int min = (T_total - tempo)/60;           // Tempo que
   falta em minutos
7
8   int Consumo_agua;
9   Consumo_agua = impulsos/2/impulso;                 // Consumo de agua
   em litros
10
11  int Consumo_energia;
12  Consumo_energia = Consumed_energy();                 //
   Consumo de energia
13
14  float custo = 0;
15  custo = (Consumo_agua*CL_agua) + (C_energia*C_energia); // custo
   da lavagem
16
17  myprintdeci(Flag_start);
18  myputchar(',');
19  myprintdeci(min);
20  myputchar(',');
21  myprintdeci(temp/10);
22  myputchar(',');
23  myprintdeci(Consumo_agua);
24  myputchar(',');
25  myprintdeci(custo);
26  myputchar(',');
27  return;
28 }

1  // rotina que implementa o ciclo de lavagem
2  void Ciclo(void){
3   static unsigned int T = 3;
4   cData = 0b00000000;
5
```

```
6     if(state == 0){
7         cData = 0b01000000;
8         SPI_Transmit(cData);
9         state = 1;          //Drenar
10        counter = 0;
11    }
12    if(state == 1 && counter >= 60){
13        cData = 0b10000000;
14        SPI_Transmit(cData);
15        state = 2;
16    }
17    if(state == 2 && !(PINB & 16)){
18        cData = 0b00000000;
19        SPI_Transmit(cData);
20        state = 3;          //Esvaziar
21    }
22    if(state == 3 && (PINB & 16) ){
23        cData = 0b10000000;
24        SPI_Transmit(cData);
25        counter = 0;
26        state = 4;
27    }
28    if(state == 4 && !(PINB & 16) ){
29        counter = 0;
30        cData = 0b00000000;
31        SPI_Transmit(cData);
32        if(T == 3){
33            state = 8;
34        }else if(T == 2){
35            state = 5;
36        }
37    }
38    if(state == 5 && !(PINB >> 4 & 1)){
39        cData = 0b01000000;
40        SPI_Transmit(cData);
41        counter = 0;
42        state = 30;
43        T = 3;
44    }
45    if(state == 30 && (PINB & 16) && counter >= 40){
46        cData = 0b10000000;
47        SPI_Transmit(cData);
48        state = 31;
49    }
50    if(state == 31 && !(PINB & 16) ){
```



```
51     cData = 0b00000000;
52     SPI_Transmit(cData);
53     state = 32;    //esvaziar
54 }
55 if(state == 32 && (PINB & 16) ){
56     cData = 0b10000000;
57     SPI_Transmit(cData);
58     state = 33;
59 }
60 if(state == 33 && !(PINB & 16) ){
61     cData = 0b00000000;
62     SPI_Transmit(cData);
63     counter = 0;
64     state = 8;    //Esvaziar A
65 }
66 if(state == 8 && !(PINB & 16)  && counter >= 10){
67     counter = 0;
68     cData = 0b00000100;
69     SPI_Transmit(cData);
70     state = 9;
71 }
72 if(state == 9 && (PINB & 16) ){
73     cData = 0b10000100;
74     SPI_Transmit(cData);
75     counter = 0;
76     state = 10;
77 }
78
79 if(state == 10 && !(PINB & 16) ){
80     counter = 0;
81     if(programa == 3){ //Programa meia carga
82         programa = 4;
83         cData = 0b00000100;
84         SPI_Transmit(cData);
85         state = 16;
86     }else if(programa == 2){ //Programa carga completa
87         cData = 0b00000100;
88         SPI_Transmit(cData);
89         programa = 3;
90         state = 14;
91     }else if(programa == 1){ //Programa carga completa intensivo
92         cData = 0b00000100;
93         SPI_Transmit(cData);
94         programa = 2;
95         state = 15;
```

```
96     }else if(programa == 0){ //Programa pre lavagem
97         cData = 0b00000100;
98         SPI_Transmit(cData);
99         state = 13;
100     }
101     else if(programa == 4){ //Parte dois depositos +3
102         cData = 0b00000100;
103         SPI_Transmit(cData);
104         programa = 5;
105         state = 0;
106     }
107     else if(programa == 5){ //Parte dos 3 depositos e final
108         cData = 0b00000100;
109         SPI_Transmit(cData);
110         state = 22;
111     }
112 }
113 if(state == 13 && counter >= 260){
114     cData = 0b00000000;
115     SPI_Transmit(cData);
116     set_initial_conditions();
117     T = 3;
118 }
119 if(state == 14 && counter >= 130){
120     state = 0;    // microciclo
121 }
122 if(state == 15 && counter >= 130){
123     state = 0;    // microciclo
124 }
125 if(state == 16 && counter >= 130){
126     cData = 0b00010100;
127     SPI_Transmit(cData);
128     counter = 0;
129     state = 17;
130 }
131 if(state == 17 && counter >= 60){
132     cData = 0b00100110;
133     SPI_Transmit(cData);
134     counter = 0;
135     state = 18;
136 }
137 if(state == 18 && counter >= 60){
138     counter = 0;
139     cData = 0b00000110;
140     SPI_Transmit(cData);
```

```
141     state = 19;
142 }
143 if(state == 19 && counter >= 1750){
144     counter = 0;
145     cData = 0b00100110;
146     SPI_Transmit(cData);
147     state = 20;
148 }
149 if(state == 20 && counter >= 60){
150     counter = 0;
151     cData = 0b00000100;
152     SPI_Transmit(cData);
153     state = 21;
154 }
155 if(state == 21 && counter >= 370){
156     T = 2;
157     counter = 0;
158     state = 0;
159 }
160 if(state == 22 && counter >= 10){
161     counter = 0;
162     cData = 0b00000110;
163     SPI_Transmit(cData);
164     state = 23;
165 }
166 if(state == 23 && counter >= 1750){
167     counter = 0;
168     cData = 0b00010100;
169     SPI_Transmit(cData);
170     state = 24;
171 }
172 if(state == 24 && counter >= 60){
173     counter = 0;
174     cData = 0b00000100;
175     SPI_Transmit(cData);
176     state = 25;
177 }
178 if(state == 25 && counter >= 60){
179     counter = 0;
180     cData = 0b00010100;
181     SPI_Transmit(cData);
182     state = 26;
183 }
184 if(state == 26 && counter >= 60){
185     counter = 0;
```

```

186     cData = 0b00000100;
187     SPI_Transmit(cData);
188     state = 27;
189 }
190 if(state == 27 && counter >= 60){
191     counter = 0;
192     cData = 0b01000000;
193     SPI_Transmit(cData);
194     state = 28;
195 }
196 if(state == 28 && counter >= 60){
197     cData = 0b00000000;
198     SPI_Transmit(cData);
199     T = 3;
200     set_initial_conditions();
201     //FIM
202 }
203 }

1 // variaveis globais usadas
2 //FLAGS
3 volatile unsigned int          Flag_start = 0;
4 volatile unsigned int          Flag_standby = 1;
5 volatile unsigned int          Flag_botao = 1;
6
7 //Sensores
8 volatile unsigned int          counter = 0;          // conta o
    tempo em cada estado
9 unsigned int                  tempo = 0;          // conta o tempo
    total da lavagem
10 unsigned long long int        impulsos = 0;          // impulsos
11 unsigned long long int        Temperatura = 70;
12
13 volatile unsigned int          state = 0;
14 unsigned int                  programa = 2; //Default carga completa
15 volatile unsigned long long int Pindeferida = 0;
16 char                          cData = 0b00000000;
17 char                          note = 48;
18
19 volatile uint8_t value = 8, old_value = 0;
20 volatile uint8_t ppa, ppo = 1;
21 // rotina principal, aqui sao chamadas todas as rotinas de
    configuracao. Tambem e feito a atualizacao do estado ciclo de
    lavagem e a atualizacao do painel local da maquina. Ou entao
    atualiza o programa de lavagem selecionado
22 int main (void){

```

```
23
24   SPI_Init();
25   init_usart();
26   init_io();
27   init_PCINT9();
28   T1_init();
29
30   while(1){
31       if(Flag_start == 1){           // corre o ciclo de lavagem e
           atualiza o painel local
32           Ciclo();
33           painel_local();
34       }else{                         // atualiza a escolha do programa
35           check_programa();
36       }
37   }
38 }
39 }
```

